University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

Computer Network 1
شبكات الحاسوب 1

Teaching by: L. Wisam Mahmood
م. وسام محمود

prepare by: Dr. Ekhlas Khalaf

cs.uotechnology.edu.iq

Congestion Control

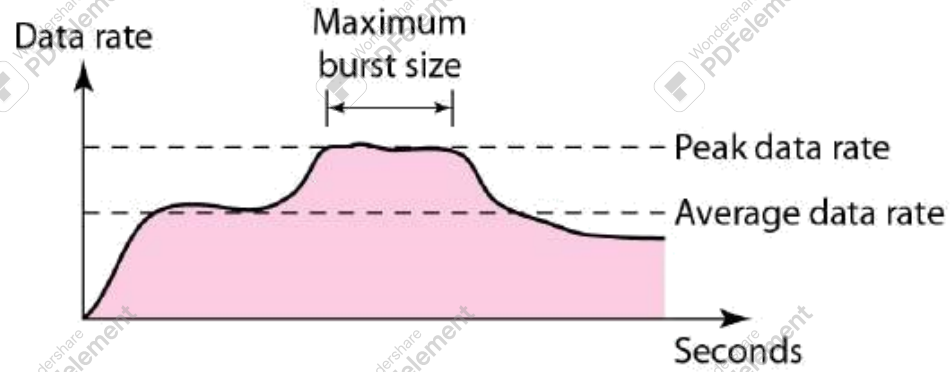
Congestion control and quality of service are two issues so closely bound together that improving one means improving the other and ignoring one usually means ignoring the other. Most techniques to prevent or eliminate congestion also improve the quality of service in a network.

■ Data traffic

The main focus of congestion control and quality of service is data traffic. In congestion control we try to avoid traffic congestion. In quality of service, we try to create an appropriate environment for the traffic.

Traffic Descriptor

Traffic descriptors are qualitative values that represent a data flow. Figure shows a traffic flow with some of these values.



Average Data Rate

The average data rate is the number of bits sent during a period of time, divided by the number of seconds in that period. We use the following equation:

Average data rate =amount of data/ Time

The average data rate is a very useful characteristic of traffic because it indicates the average bandwidth needed by the traffic.

Peak Data Rate

The peak data rate defines the maximum data rate of the traffic. In Figure it is the maximum y axis value. The peak data rate is a very important measurement because it indicates the peak bandwidth that the network needs for traffic to pass through without changing its data flow.

Maximum Burst Size

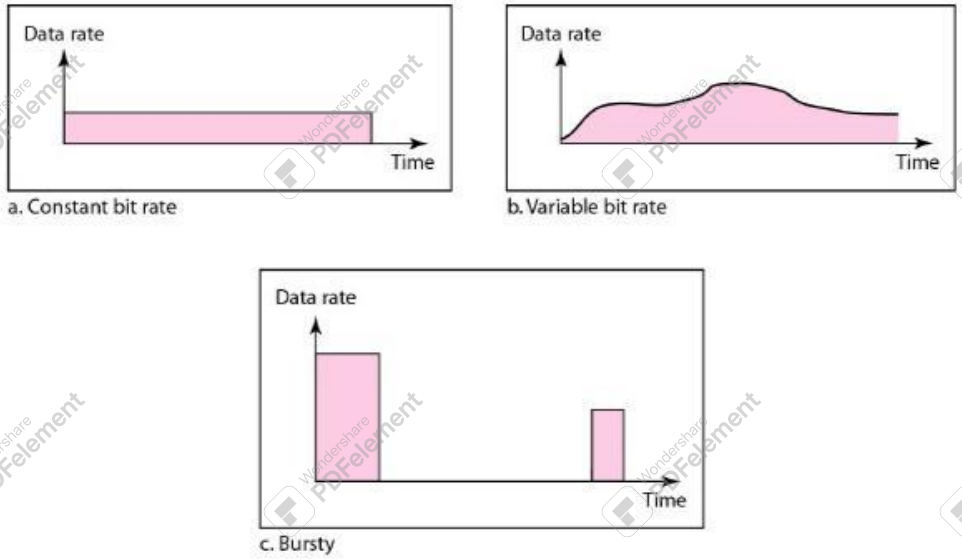
Although the peak data rate is a critical value for the network, it can usually be ignored if the duration of the peak value is very short. For example, if data are flowing steadily at the rate of 1 Mbps with a sudden peak data rate of 2 Mbps for just 1 ms, the network probably can handle the situation. However, if the peak data rate lasts 60 ms, there may be a problem for the network. The maximum burst size normally refers to the maximum length of time the traffic is generated at the peak rate.

Effective Bandwidth

The effective bandwidth is the bandwidth that the network needs to allocate for the flow of traffic. The effective bandwidth is a function of three values: average data rate, peak data rate, and maximum burst size. The calculation of this value is very complex.

Traffic Profiles

For our purposes, a data flow can have one of the following traffic profiles: constant bit rate, variable bit rate, or bursty as shown in Figure.



Constant Bit Rate

A constant-bit-rate (CBR), or a fixed-rate, traffic model has a data rate that does not change. In this type of flow, the average data ratc and thc peak data rate are the same. The maximum burst size is not applicable. This type of traffic is very easy for a network to handle since it is predictable. The network knows in advance how much bandwidth to allocate for this type of flow.

Variable Bit Rate

In the variable -bit-rate (VBR) category, the rate of the data flow changes in time, with the changes smooth instead of sudden and sharp. In this type of flow, the average data rate and the peak data rate are different.

Bursty

In the **bursty data** category, the data rate changes suddenly in a very short time. It may jump from zero, for example, to 1 Mbps in a few microseconds and vice versa. It may also remain at this value for a while. The average bit rate and the

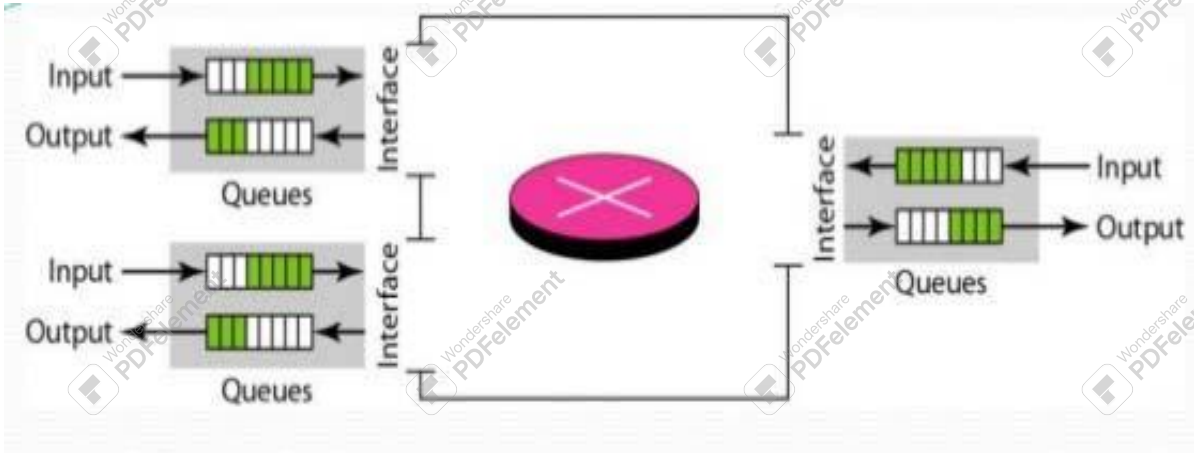
peak bit rate are very different values in this type of flow. The maximum burst size is significant. This is the most difficult type of traffic for a network to handle because the profile is very unpredictable. To handle this type of traffic, the network normally needs to reshape it, using reshaping techniques. Bursty traffic is one of the main causes of congestion in a network.

■ Congestion

An important issue in a packet-switched network is **congestion.** Congestion in a network may occur if the **load** on the network-the number of packets sent to the network-is greater than the *capacity* of the network-the number of packets a network can handle.

Congestion control refers to the mechanisms and techniques to control the congestion and keep the load below the capacity. We may ask why there is congestion on a network. Congestion happens in any system that involves waiting. Congestion in a network occurs because routers and switches have queues-buffers that hold the packets before and after processing. A router, for example, has an input queue and an output queue for each interface. When a

packet arrives at the incoming interface, it undergoes three steps before departing, as shown in Figure.

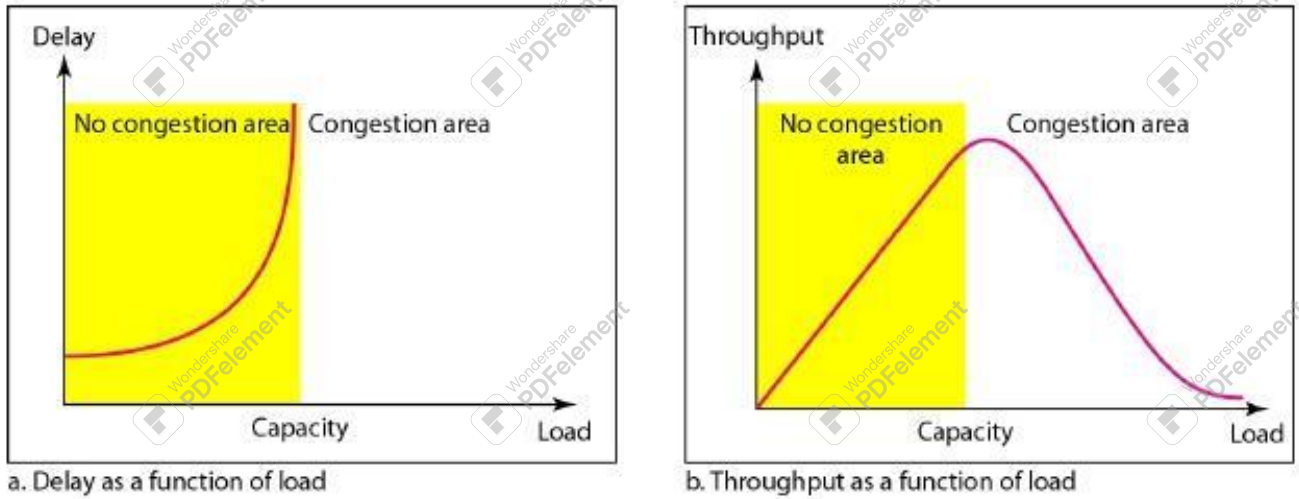


1. The packet is put at the end of the input queue while waiting to be checked.
2. The processing module of the router removes the packet from the input queue once it reaches the front of the queue and uses its routing table and the destination address to find the route.
3. The packet is put in the appropriate output queue and waits to be sent.

We need to be aware of two issues. First, if the rate of packet arrival is higher than the packet processing rate, the input queues become longer and longer. Second, if the packet departure rate is less than the packet processing rate, the output queues become longer and longer.

■ Network Performance

Congestion control involves two factors that measure the performance of a network: **delay** and **throughput**. Figure shows these two performance measures as function of load.



a. Delay as a function of load

b. Throughput as a function of load

Delay versus Load

When the load is much less than the capacity of the network, the delay is at a minimum. This minimum delay is composed of propagation delay and

processing delay, both of which are negligible. However, when the load reaches the network capacity, the delay increases sharply because we now need to add the waiting time in the queues (for all routers in the path) to the total delay. Note that the delay becomes infinite when the load is greater than the capacity. If this is not obvious, consider the size of the queues when almost no packet reaches the destination, or reaches the destination with infinite delay; the queues become longer and longer. Delay has a negative effect on the load and consequently the congestion. When a packet is delayed, the source, not receiving the acknowledgment, retransmits the packet, which makes the delay, and the congestion, worse.

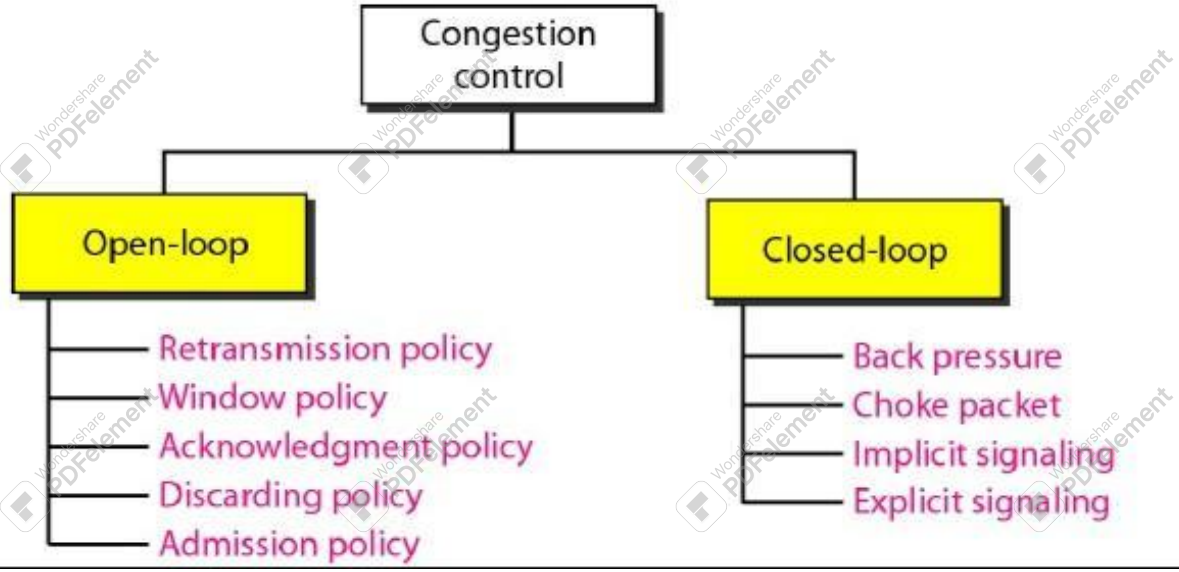
Throughput versus Load

We can define throughput in a network as the number of packets passing through the network in a unit of time. Notice that when the load is below the capacity of the network, the throughput increases proportionally with the load. We expect the throughput to remain constant after the load reaches the capacity, but instead the throughput declines sharply. The reason is the discarding of packets by the routers. When the load exceeds the capacity, the queues become full and the routers have to discard some packets. Discarding packet does not

reduce the number of packets in the network because the sources retransmit the packets, using time-out mechanisms, when the packets do not reach the destinations.

■ CONGESTION CONTROL

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. In general, we can divide congestion control mechanisms into two broad categories: open-loop congestion control (prevention) and closed-loop congestion control (removal) as shown in Figure.



Open-Loop Congestion Control

In open-loop congestion control, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination. We give a brief list of policies that can prevent congestion.

Retransmission Policy

Retransmission is sometimes unavoidable. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. Retransmission in general may increase congestion in the network. However, a good retransmission policy can prevent congestion. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion. For example, the retransmission policy used by TCP (explained later) is designed to prevent or alleviate congestion.

Window Policy

The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control. In

the Go- Back-N window, when the timer for a packet times out, several packets may be resent, although some may have arrived safe and sound at the receiver. This duplication may make the congestion worse. The Selective Repeat window, on the other hand, tries to send the specific packets that have been lost or corrupted.

Acknowledgment Policy

The acknowledgment policy imposed by the receiver may also affect congestion. If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion. Several approaches are used in this case. A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires. A receiver may decide to acknowledge only N packets at a time. We need to know that the acknowledgments are also part of the load in a network. Sending fewer acknowledgments means imposing less load on the network.

Discarding Policy

A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission.

Admission Policy

An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual- circuit networks. Switches in a flow first check the resource requirement of a flow before admitting it to the network. A router can deny establishing a virtual circuit connection if there is congestion in the network or if there is a possibility of future congestion.

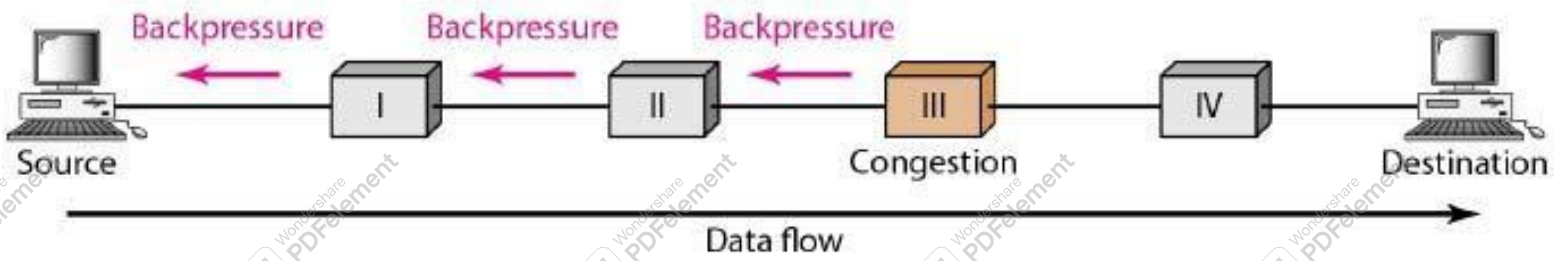
Closed-Loop Congestion Control

Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols. We describe a few of them here.

Backpressure

The technique of backpressure refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes to become congested, and

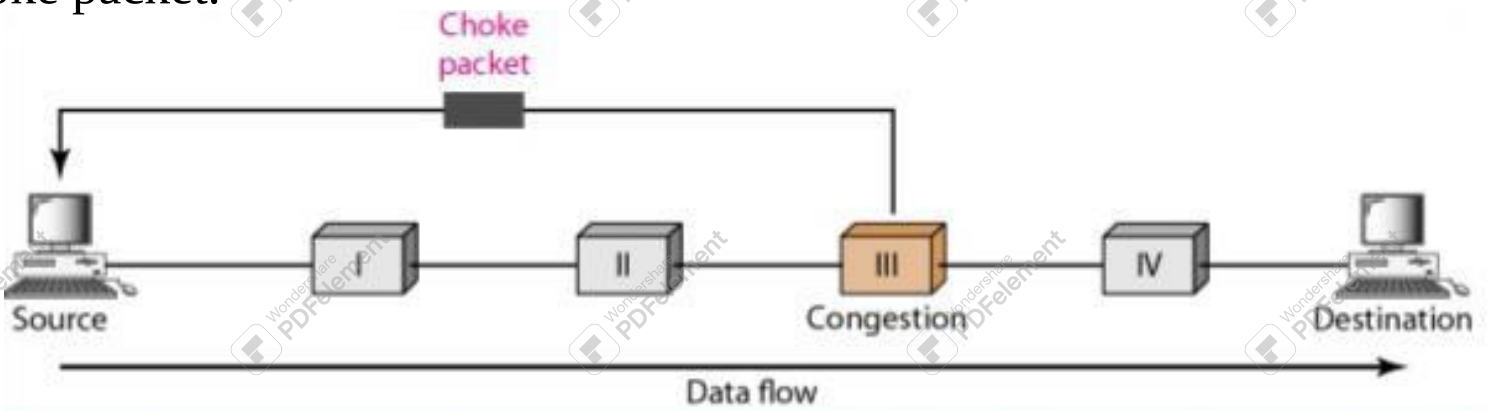
they, in turn, reject data from their upstream nodes or nodes. And so on. Figure shows the idea of backpressure.



Choke Packet

A choke packet is a packet sent by a node to the source to inform it of congestion. Note the difference between the backpressure and choke packet methods. In backpressure, the warning is from one node to its upstream node, although the warning may eventually reach the source station. In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has traveled are not warned. We have seen an example of this type of

control in ICMP. The warning message goes directly to the source station; the intermediate routers, and does not take any action. Figure shows the idea of a choke packet.



Implicit Signaling

In implicit signaling, there is no communication between the congested node or nodes and the source. The source guesses that there is congestion somewhere in the network from other symptoms. For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested.

Explicit Signaling

The node that experiences congestion can explicitly send a signal to the source or destination. The explicit signaling method, however, is different from the choke packet method. In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method, the signal is included in the packets that carry data. Explicit signaling, as we will see in Frame Relay congestion control, can occur in either the forward or the backward direction.

Backward Signaling

A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.

Forward Signaling

A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can

use policies, such as slowing down the acknowledgments, to alleviate the congestion.

TWO EXAMPLES

To better understand the concept of congestion control, let us give two examples: one in TCP and the other in Frame Relay.

Congestion Control in TCP

We now show how TCP uses congestion control to avoid congestion or alleviate congestion in the network.

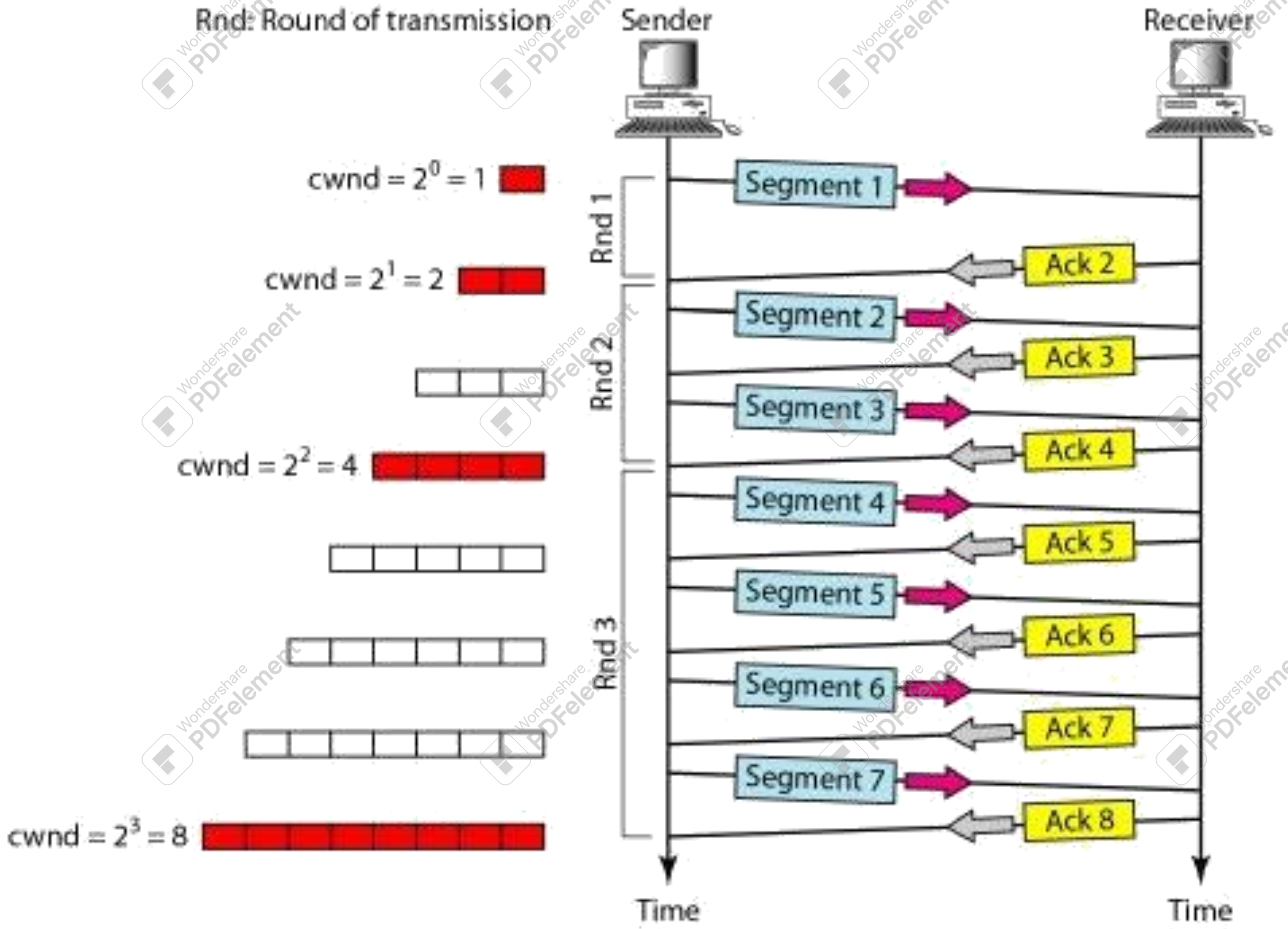
Congestion Window

Today, the sender's window size is determined not only by the receiver but also by congestion in the network. The sender has two pieces of information: the receiver-advertised window size **(rwnd)** and the congestion window size **(cwnd)**. The actual size of the window is the minimum of these two.

Actual window size= minimum (rwnd, cwnd)

Congestion Policy

TCP's general policy for handling congestion is based on three phases: slow start, congestion avoidance, and congestion detection. In the slow-start phase, the sender starts with a very slow rate of transmission, but increases the rate rapidly to reach a threshold. When the threshold is reached, the data rate is reduced to avoid congestion.
Finally if congestion is detected, the sender goes back to the slow-start or congestion avoidance phase based on how the congestion is detected.
Slow Start: Exponential Increase One of the algorithms used in TCP congestion control is called slow start.
This algorithm is based on the idea that the size of the congestion window (cwnd) starts with one maximum segment size (MSS). The MSS is determined during connection establishment by using an option of the same name. The size of the window increases one MSS each time an acknowledgment is received. As the name implies, the window starts slowly, but grows exponentially. To show the idea, let us look at Figure. Note that we have used three simplifications to make the discussion more understandable.



We have used segment numbers instead of byte numbers (as though each segment contains only 1 byte). We have assumed that rwnd is much higher than

cwnd, so that the sender window size always equals cwnd. We have assumed that each segment is acknowledged individually.

The sender starts with cwnd =1 MSS. This means that the sender can send only one segment. After receipt of the acknowledgment for segment 1, the size of the congestion window is increased by 1, which means that cwnd is now 2. Now two more segments can be sent. When each acknowledgment is received, the size of the window is increased by 1 MSS. When all seven segments are acknowledged, cwnd = 8.

If we look at the size of *cwnd* in terms of rounds (acknowledgment of the whole window of segments), we find that the rate is exponential as shown below:

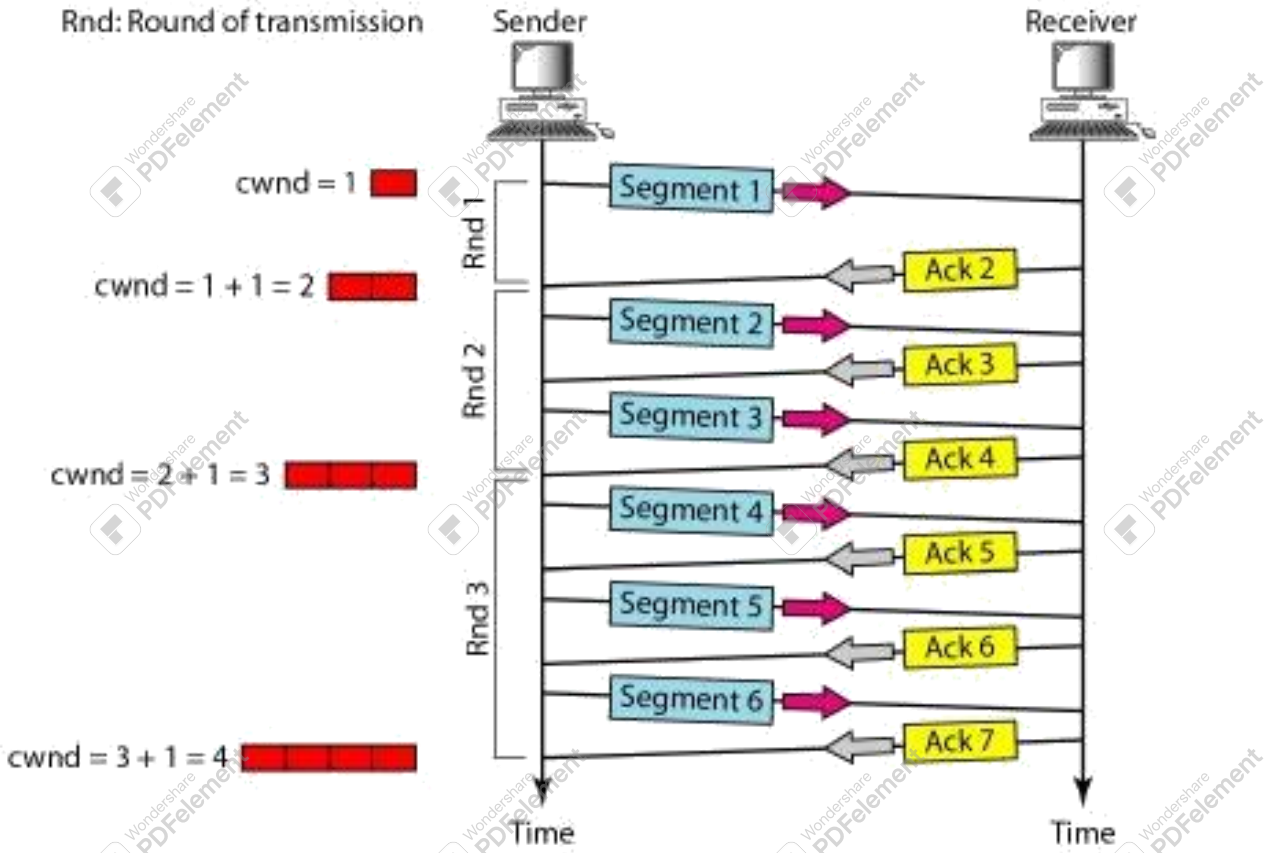
$$\text{Start} \rightarrow \text{cwnd}=1$$
$$\text{After round 1} \rightarrow \text{cwnd}=2^1=2$$
$$\text{After round 2} \rightarrow \text{cwnd}=2^2=4$$
$$\text{After round 3} \rightarrow \text{cwnd}=2^3=8$$

There must be a threshold to stop this phase. The sender keeps track of a variable named ssthresh (slow-start threshold). When the size of window in bytes reaches this threshold, slow start stops and the next phase starts. In most implementations the value of ssthresh is 65,535 bytes.

Congestion Avoidance: Additive Increase If we start with the slow-start algorithm, the size of the congestion window increases exponentially. To avoid congestion before it happens, one must slow down this exponential growth. TCP defines another algorithm called congestion avoidance, which undergoes an additive increase instead of an exponential one.

When the size of the congestion window reaches the slow-start threshold, the slow-start phase stops and the additive phase begins. In this algorithm, each time the whole window of segments is acknowledged (one round), the size of the congestion window is increased by 1.

To show the idea, we apply this algorithm to the same scenario as slow start, although we will see that the congestion avoidance algorithm usually starts when the size of the window is much greater than 1. Figure shows the idea.



In this case, after the sender has received acknowledgments for a complete window size of segments, the size of the window is increased by one segment. If we look at the size of cwnd in terms of rounds, we find that the rate is additive as shown below:

Start → cwnd=l
After round 1 → cwnd= 1+ 1 =2
After round 2 → cwnd=2+ 1 =3
After round 3 → cwnd=3+ 1 =4

Congestion Detection: Multiplicative Decrease If congestion occurs, the congestion window size must be decreased. The only way the sender can guess that congestion has occurred is by the need to retransmit a segment. However, retransmission can occur in one of two cases: when a timer times out or when three ACKs are received. In both cases, the size of the threshold is dropped to one-half, a multiplicative decrease. Most TCP implementations have two reactions:

1. If a time-out occurs, there is a stronger possibility of congestion; a segment has probably been dropped in the network, and there is no news about the sent segments.

   In this case TCP reacts strongly:

   a. It sets the value of the threshold to one-half of the current window size.
   b. It sets cwnd to the size of one segment.
   c. It starts the slow-start phase again.

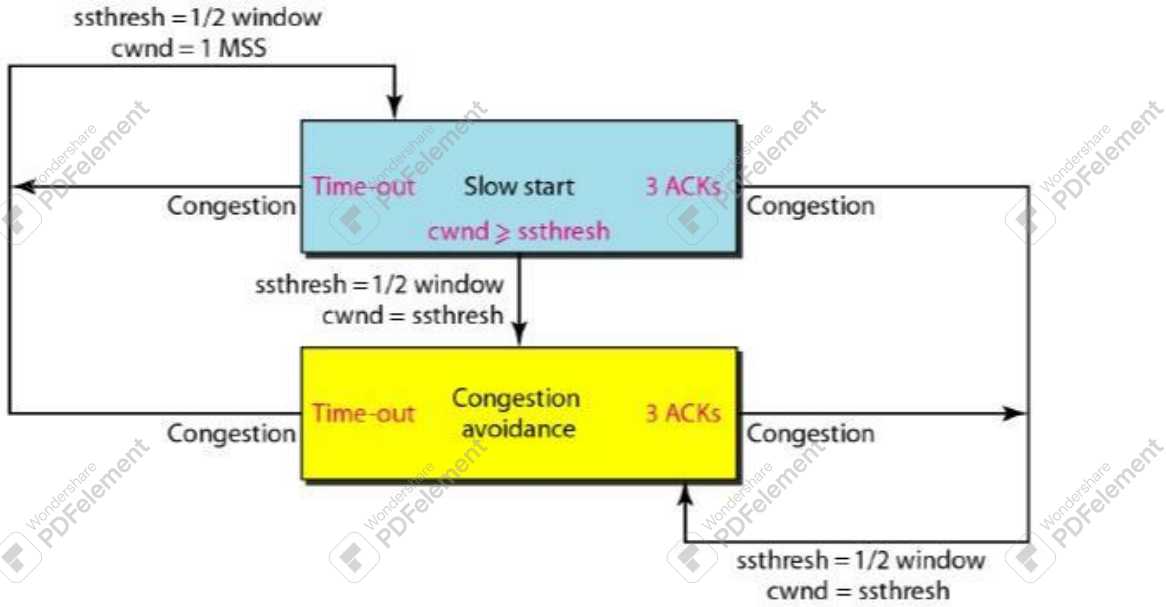
2. If three ACKs are received, there is a weaker possibility of congestion; a segment may have been dropped, but some segments after that may have arrived safely since three ACKs are received. This is called fast transmission and fast recovery. In this case, TCP has a weaker reaction:

a. It sets the value of the threshold to one-half of the current window size.

b. It sets *cwnd* to the value of the threshold (some implementations add three segment sizes to the threshold).

c. It starts the congestion avoidance phase.

Summary in Figure, we summarize the congestion policy of TCP and the relationships between the three phases.

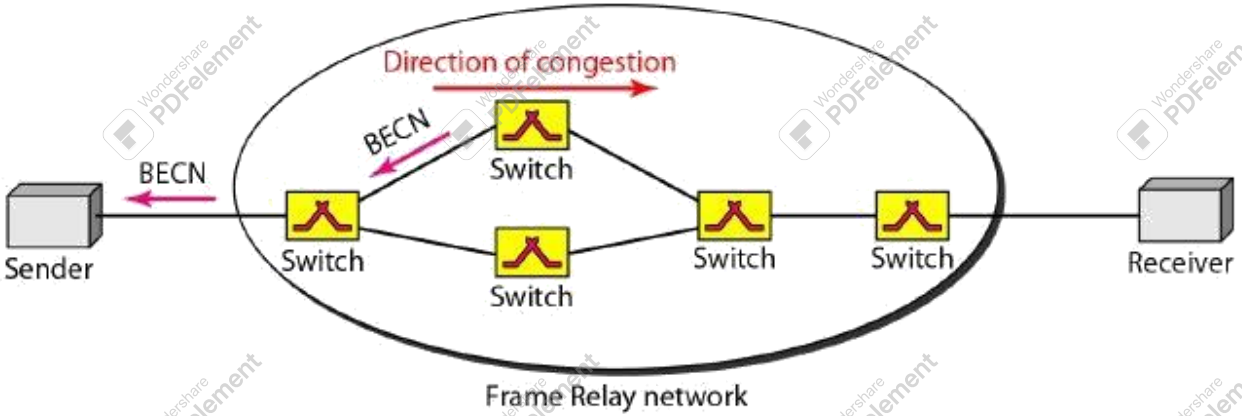


Congestion Control in Frame Relay

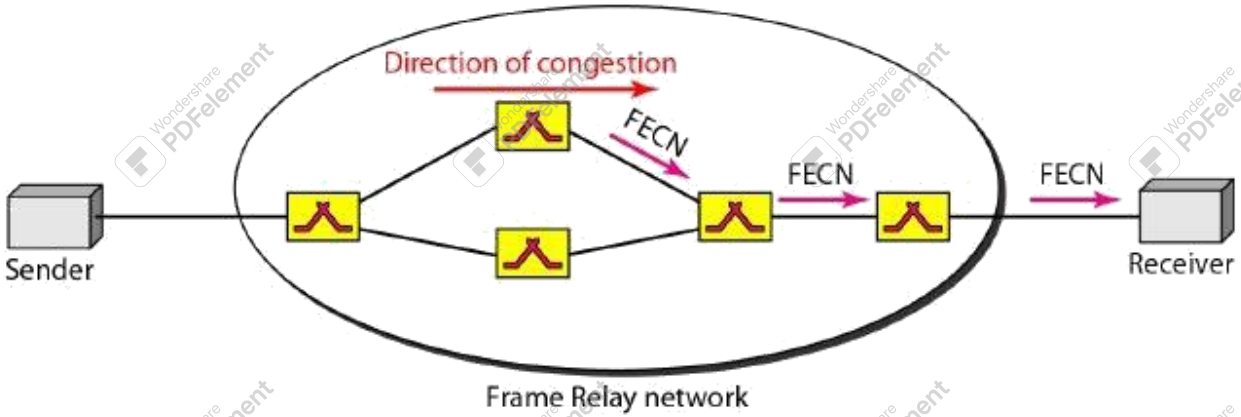
Congestion in a Frame Relay network decreases throughput and increases delay. A high throughput and low delay are the main goals of the Frame Relay protocol. Frame Relay does not have flow control. In addition, Frame Relay allows the user to transmit bursty data. This means that a Frame Relay network has the potential to be really congested with traffic, thus requiring congestion control.

Congestion Avoidance

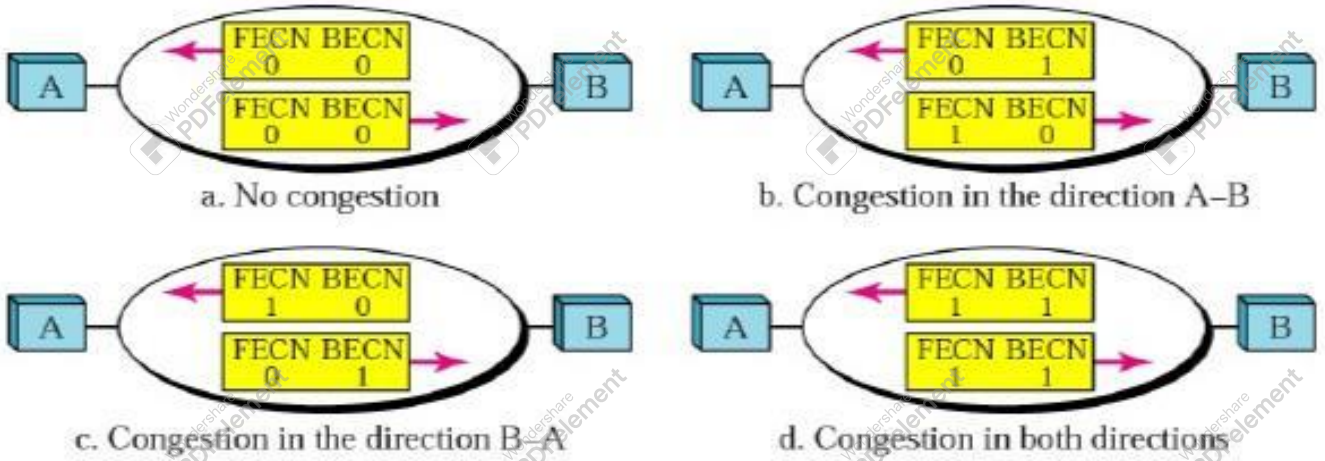
For congestion avoidance, the Frame Relay protocol uses 2 bits in the frame to explicitly warn the source and the destination of the presence of congestion. BECN The backward explicit congestion notification **(BECN)** bit warns the sender of congestion in the network. In fact, there are two methods: The switch can use response frames from the receiver (full- duplex mode), or else the switch can use a predefined connection (DLCI =1023) to send special frames for this specific purpose. The sender can respond to this warning by simply reducing the data rate. Figure shows the use of BECN.



FECN The forward explicit congestion notification **(FECN)** bit is used to warn the receiver of congestion in the network. It might appear that the receiver cannot do anything to relieve the congestion. However, the Frame Relay protocol assumes that the sender and receiver are communicating with each other and are using some type of flow control at a higher level. For example, if there is an acknowledgment mechanism at this higher level, the receiver can delay the acknowledgment, thus forcing the sender to slow down. Figure shows the use of FECN.



When two endpoints are communicating using a Frame Relay network, four situations may occur with regard to congestion. Figure shows these four situations and the values of FECN and BECN.



Network Management

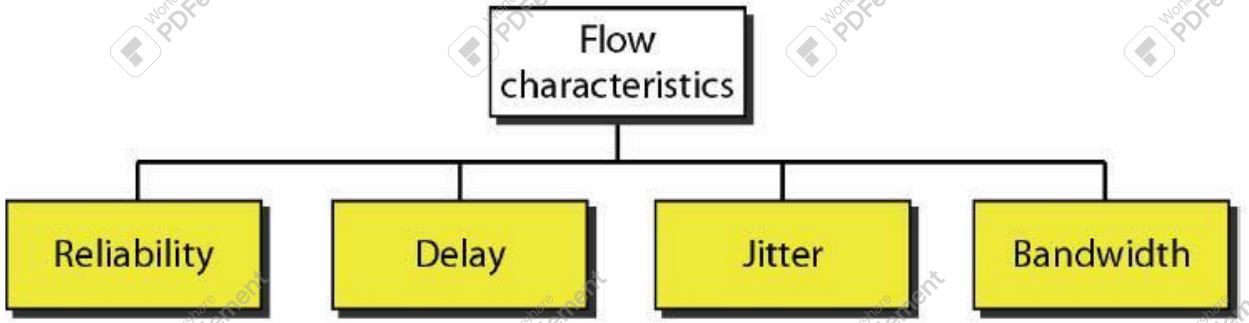
4. Quality of service

Quality of service

We can informally define quality of service as something a flow seeks to attain.
QOS relates to the kind of service a user gets from the network.
e.g., high /low bandwidth, delay, loss.
Quality of Service (QoS) denotes the well defined and manageable behavior of a system according to measurable parameters.

Flow Characteristics

Traditionally, four types of characteristics are attributed to a flow:
reliability, delay, jitter, and bandwidth, as shown in Figure.



Reliability

Reliability is a characteristic that a flow needs. Reliability means: delivery of packets without loss, duplication and error. Lack of reliability means: losing a packet or acknowledgement. Applications have different reliability needs

- Email, file transfer and internet access require more reliability

- Telephony and audio conferencing requires less reliability

Delay

End-to-end delay: the time required for a packet to travel from source to destination. Different applications have different delay needs

-Telephony, audio conferencing, video conferencing and remote login requires minimum delay.

- Delay of file transfer or email is not so important.

Jitter

Jitter is defined as the variation in the packet delay belonging to the same flow. High jitter means the difference between delays is large; low jitter means the variation is small.

Example

Packets	Departing time	Arrival time	Delay (s)
Packet 1	10:00	10:05	5
Packet 2	10:01	10:08	7
Packet 3	10:02	10:08	6
Packet 4	10:03	10:11	8

different applications have different jitter needs

- for audio and video the above scenario is not acceptable
- for emails and file transfer the above scenario is acceptable

Bandwidth

Bandwidth is the band (range) of data transfer rate. Different applications need different bandwidths. Video conferencing requires more bandwidth but email requires less bandwidth. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million.

Flow Classes

Based on the flow characteristics, we can classify flows into groups, with each group having similar levels of characteristics. This categorization is not formal or universal; some protocols such as ATM have defined classes, as we will see later.

Techniques to improve QOS

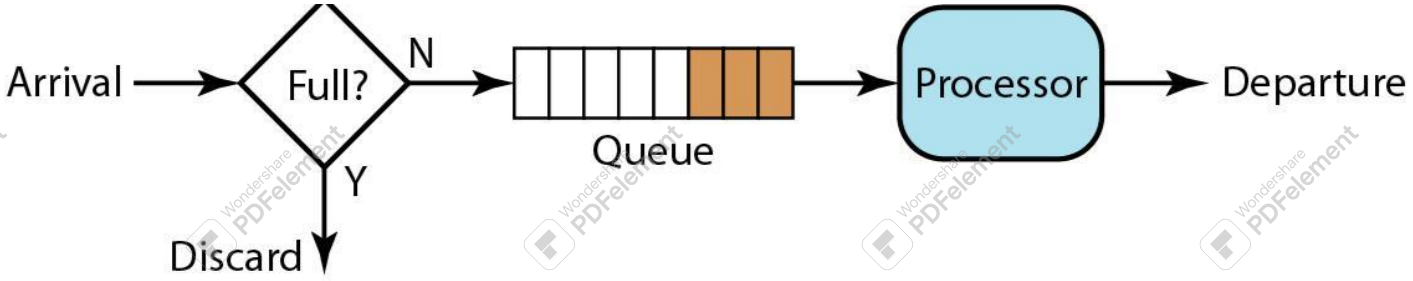
We briefly discuss four common methods: scheduling, traffic shaping, admission control, and resource reservation.

1. Scheduling

Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Several scheduling techniques are designed to improve the quality of service. We discuss three of them here: FIFO queuing, priority queuing, and weighted fair queuing.

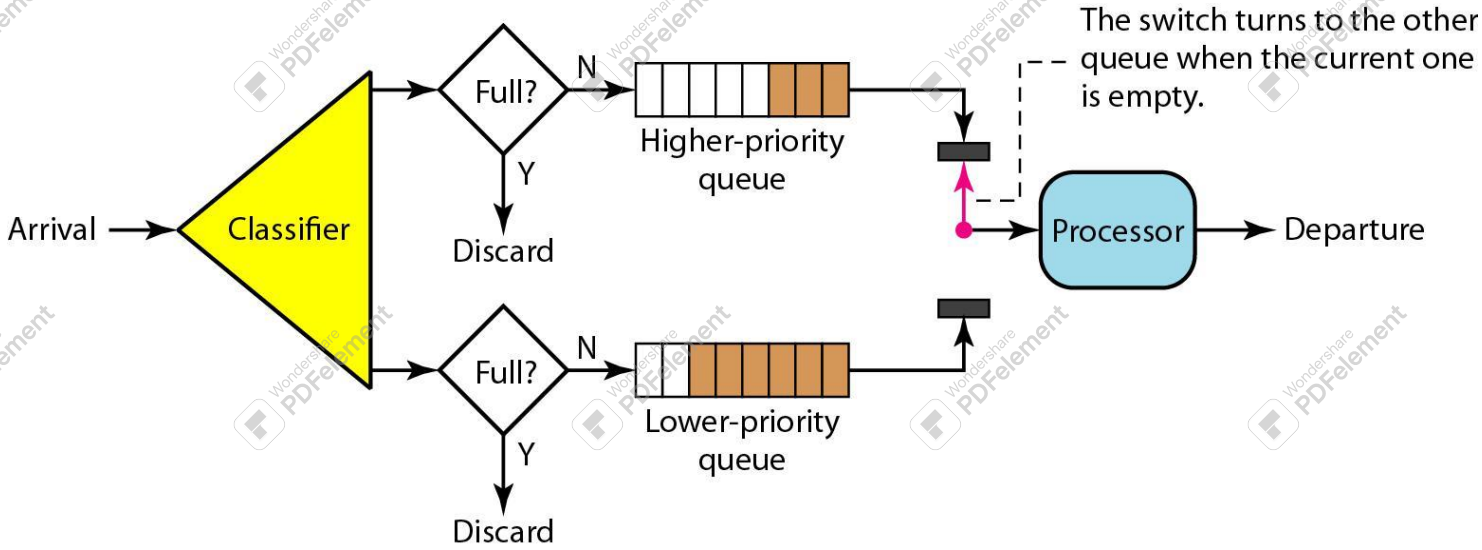
FIFO Queuing

In first-in, first-out (FIFO) queuing, packets wait in a buffer (queue) until the node (router or switch) is ready to process them. If the average arrival rate is higher than the average processing rate, the queue will fill up and new packets will be discarded. A FIFO queue is familiar to those who have had to wait for a bus at a bus stop. Figure shows a conceptual view of a FIFO queue.



Priority Queuing

In priority queuing, packets are first assigned to a priority class. Each priority class has its own queue. The packets in the highest-priority queue are processed first. Packets in the lowest-priority queue are processed last. Note that the system does not stop serving a queue until it is empty. Figure shows priority queuing with two priority levels (for simplicity).

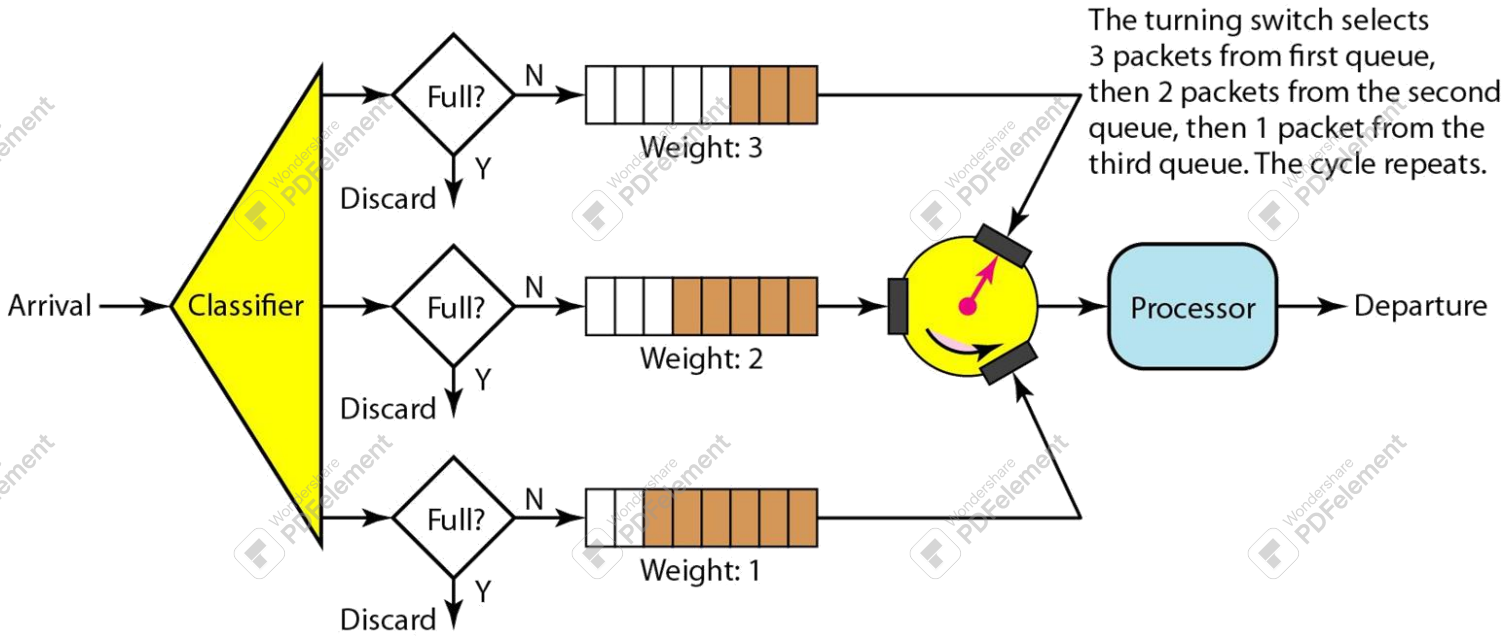


A priority queue can provide better QoS than the FIFO queue because higher priority traffic, such as multimedia, can reach the destination with less delay. However, there is a potential drawback. If there is a continuous flow in a high-

priority queue, the packets in the lower-priority queues will never have a chance to be processed. This is a condition called **starvation**.

Weighted Fair Queuing

A better scheduling method is weighted fair queuing. In this technique, the packets are still assigned to different classes and admitted to different queues. The queues, however, are weighted based on the priority of the queues; higher priority means a higher weight. The system processes packets in each queue in a round- robin fashion with the number of packets selected from each queue based on the corresponding weight. For example, if the weights are 3, 2, and 1, three packets are processed from the first queue, two from the second queue, and one from the third queue. If the system does not impose priority on the classes, all weights can be equal in this way, we have fair queuing with priority. Figure shows the technique with three classes.



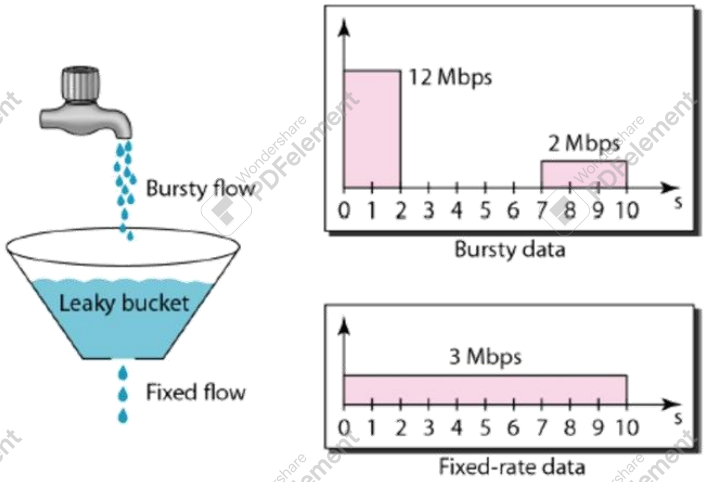
2. Traffic Shaping

Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic: leaky bucket and token bucket.

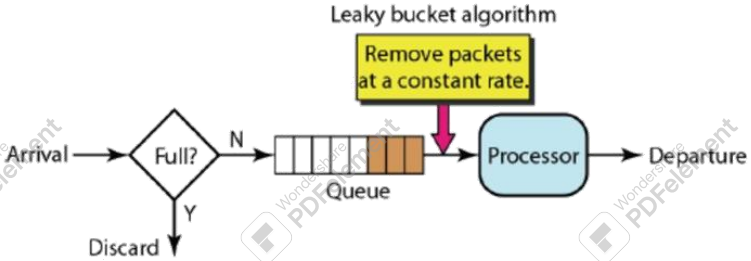
Leaky Bucket

If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket. The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty. The input rate can vary, but the output rate remains constant. Similarly, in networking, a technique called **leaky bucket** can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate.

In the figure, we assume that the network has committed a bandwidth of 3 Mbps for a host. The use of the leaky bucket shapes the input traffic to make it conform to this commitment. The host sends a burst of data at a rate of 12 Mbps for 2 s, for a total of 24 Mbits of data. The host is silent for 5 s and then sends data at a rate of 2 Mbps for 3 s, for a total of 6 Mbits of data. In all, the host has sent 30 Mbits of data in 10s.
The leaky bucket smooths the traffic by sending out data at a rate of 3 Mbps during the same 10 s. Without the leaky bucket, the beginning burst may have hurt the network by consuming more bandwidth than is set aside for this host. We can also see that the leaky bucket may prevent congestion.



- Drop the packet if the bucket is full

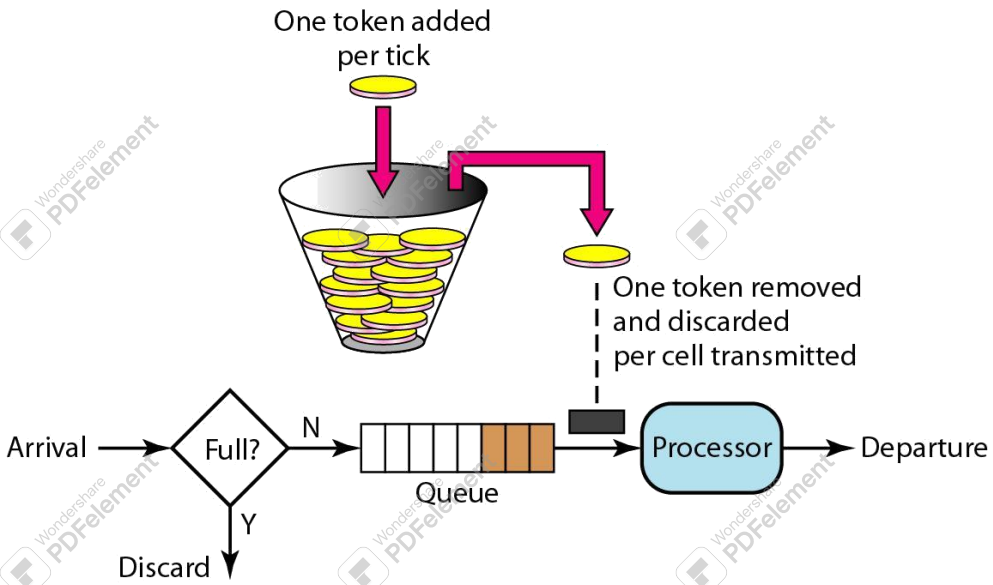


A leaky bucket algorithm shapes bursty traffic into fixed-rate traffic by averaging the data rate. It may drop the packets if the bucket is full.

Token Bucket

The leaky bucket is very restrictive. It does not credit an idle host. For example, if a host is not sending for a while, its bucket becomes empty. Now if the host has bursty data, the leaky bucket allows only an average rate. The time when the host was idle is not taken into account. On the other hand, the token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens. For each tick of the clock, the system sends **n** tokens to the bucket. The system removes one token for every cell (or byte) of data sent. For example, if **n** is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens. Now the host can consume all these tokens in one tick with 10,000 cells, or the host takes 1000 ticks with 10 cells per tick. In other words, the host can send bursty data as long as the bucket is not empty. Figure shows the idea.

The token bucket allows bursty traffic at a regulated maximum rate.



The token bucket can easily be implemented with a counter. The token is initialized to zero. Each time a token is added, the counter is incremented by 1. Each time a unit of data is sent, the counter is decremented by 1. When the counter is zero, the host cannot send data.

Combining Token Bucket and Leaky Bucket

The two techniques can be combined to credit an idle host and at the same time regulate the traffic. The leaky bucket is applied after the token bucket; the rate of the leaky bucket needs to be higher than the rate of tokens dropped in the bucket.

3. Admission Control

Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

4. Resource Reservation

A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. The quality of service is improved if these resources are reserved beforehand. One QoS model called Integrated Services, which depends heavily on resource reservation to improve the quality of service.

Network Management

5. Integrated & differentiated Services

Integrated Services

Two models have been designed to provide quality of service in the Internet: Integrated Services and Differentiated Services. Both models emphasize the use of quality of service at the network layer (IP), although the model can also be used in other layers such as the data link.
IP was originally designed for best-effort delivery. This means that every user receives the same level of services. This type of delivery does not guarantee the minimum of a service, such as bandwidth, to applications such as real-time audio and video. If such an application accidentally gets extra bandwidth, it may be detrimental to other applications, resulting in congestion.

Integrated Services (IntServ) is a reservation based model. The intention is to guarantee individual QoS profiles for each flow.

Integrated Services is a flow-based QoS model designed for IP.

What is a flow?

- A flow is a stream of packets originated from the same application session
- The term "flow" describes semantically coherence of data

Categories of applications

- Elastic applications, no delivery requirements as long as the packets reach the destination, e.g. TCP traffic (machine to machine)
- Real Time Tolerant (RTT) applications, demand weak bounds for the maximum transfer delay, also some packet loss is acceptable, e.g. streamed video (machine to human)
- Real Time Intolerant (RTI) applications, demand minimal delay and jitter, e.g. interactive application or videoconferences (human to human)

■ Signaling

IP is a connectionless, datagram, packet-switching protocol.

How can we implement a flow-based model over a connectionless protocol?

The solution is a signaling protocol to run over IP that provides the signaling mechanism for making a reservation. This protocol is called Resource Reservation Protocol (RSVP).

■ Flow Specification

When a source makes a reservation, it needs to define a flow specification. A flow specification has two parts: **Rspec** (resource specification) and **Tspec** (traffic specification). Rspec defines the resource that the flow needs to reserve (buffer, bandwidth, etc.). Tspec defines the traffic characterization of the flow.

■ Admission

After a router receives the flow specification from an application, it decides to admit or deny the service. The decision is based on the previous commitments of the router and the current availability of the resource.

■ Service Classes

Guaranteed Service for RTI applications

This type of service is designed for real-time traffic that needs a guaranteed minimum end-to-end delay. The end -to-end delay is the sum of the delays in the routers, the propagation delay in the media, and the setup mechanism. Only the first, the sum of the delays in the routers, can be guaranteed by the router. This type of service guarantees that the packets will arrive within a certain delivery time and are not discarded if flow traffic stays within the boundary of Tspec. We can say that guaranteed services are quantitative services, in which the amount of end-to-end delay and the data rate must be defined by the application.

Controlled-Load Service for RTT applications

This type of service is designed for applications that can accept some delays, but are sensitive to an overloaded network and to the danger of losing packets. Good examples of these types of applications are file transfer, e- mail, and Internet access. The controlled load service is a qualitative type of service in that the application requests the possibility of low-loss or no-loss packets.

Best Effort Service for all other applications standard use of IP

RSVP

In the Integrated Services model, an application program needs resource reservation. As we learned in the discussion of the IntServ model, the resource reservation is for a flow. This means that if we want to use IntServ at the IP level, we need to create a flow, a kind of virtual-circuit network, out of the IP, which was originally designed as a datagram packet-switched network. A virtual -circuit network needs a signaling system to set up the virtual circuit before data traffic can start. The Resource Reservation Protocol (RSVP) is a signaling protocol to help IP create a flow and consequently make a resource reservation.

RSVP is a general signaling protocol for QoS control services.

A main focus of RSVP is to support multicast communication

- Unicasts are treated as special cases of multicast only
- RSVP performs receiver oriented reservations

- Support different requirements of many receivers
- Support heterogeneous networks

Merging of reservation supports multiple senders in a multicast environment

Multicast Trees

RSVP is different from some other signaling systems in that it is a signaling system designed for multicasting. However, RSVP can be also used for unicasting because unicasting is just a special case of multicasting with only one member in the multicast group. The reason for this design is to enable RSVP to provide resource reservations for all kinds of traffic including multimedia which often uses multicasting.

Receiver-Based Reservation

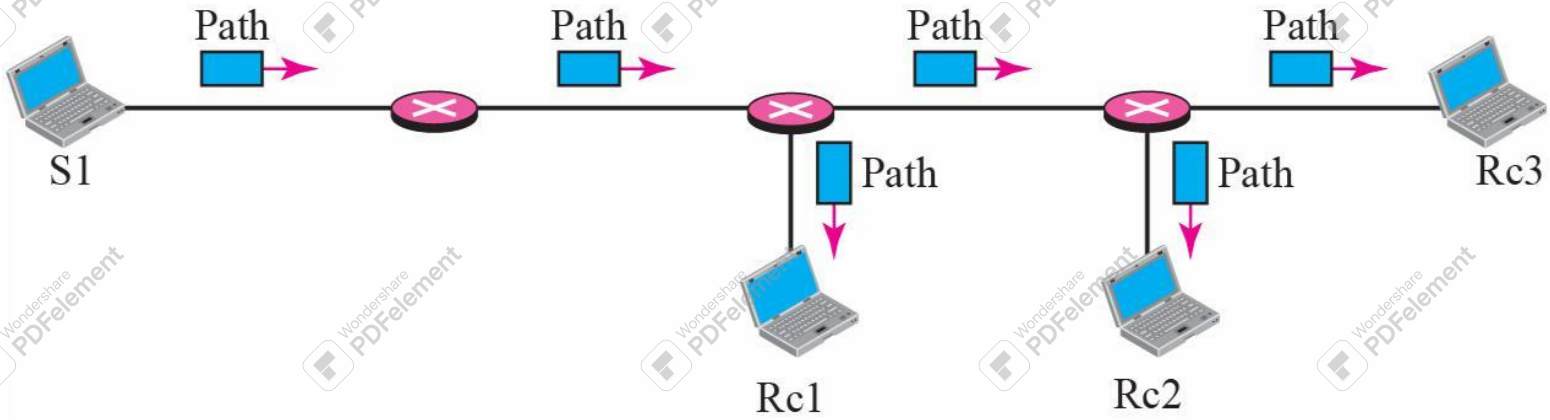
In RSVP, the receivers, not the sender, make the reservation. This strategy matches the other multicasting protocols. For example, in multicast routing protocols, the receivers, not the sender, make a decision to join or leave a multicast group.

RSVP Messages

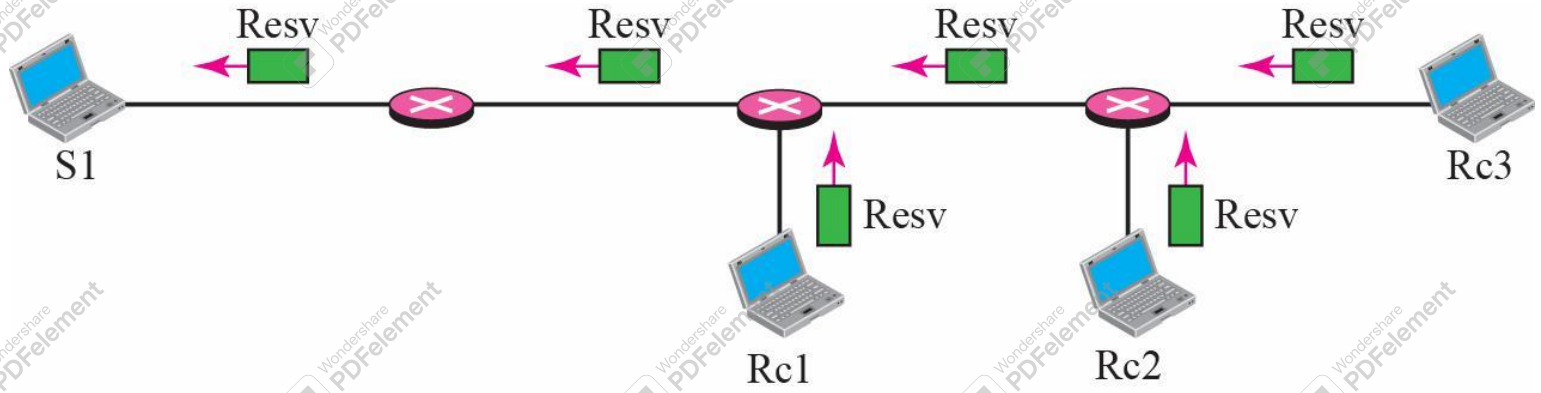
RSVP has several types of messages. We discuss only two of them: Path and Resv.

Path Messages recall that the receivers in a flow make the reservation in RSVP. However, the receivers do not know the path traveled by packets before the reservation is made. The path is needed for the reservation. To solve the problem, RSVP uses Path messages.

A Path message travels from the sender and reaches all receivers in the multicast path. On the way, a Path message stores the necessary information for the receivers. A Path message is sent in a multicast environment; a new message is created when the path diverges. Figure shows path messages.



Resv Messages After a receiver has received a Path message, it sends a Resv message. The Resv message travels toward the sender (upstream) and makes a resource reservation on the routers that support RSVP. If a router does not support RSVP on the path, it routes the packet based on the best-effort delivery methods. Figure shows the Resv messages.



A PATH message is sent from sender to receiver

- The sender specifies its traffic specification (sender TSpec)
- The sender specifies its traffic characteristic (ADSPEC)
- Detection of path characteristics

Detected bandwidth limitations, minimum packet size (MTU), may modify ADSPEC

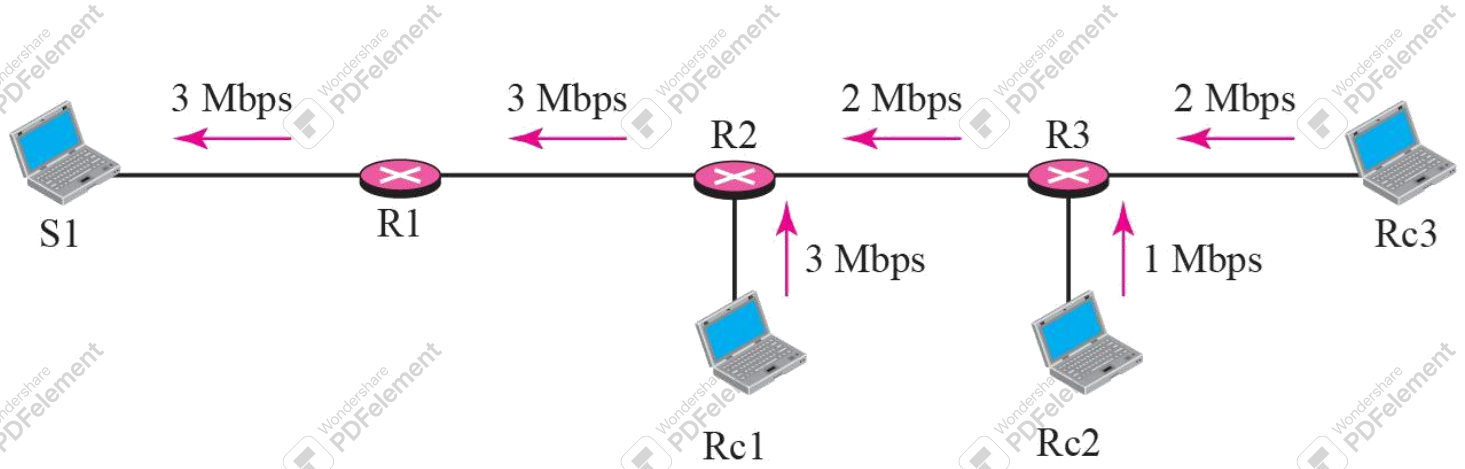
- RSVP capable nodes get to know their RSVP capable neighbors
  - RSVP does not perform routing; routing is done by standard components which do not know anything about QoS.

A RESV message is sent from receiver to sender

- The RESV message travels the path backward, perform reservations
- The receiver application determines the required resource reservation and replies with
- Traffic specification (receiver Tspec)
- Requested Service Specification (receiver Rspec)

Reservation Merging

In RSVP, the resources are not reserved for each receiver in a flow; the reservation is merged. In Figure, Rc3 requests a 2-Mbps bandwidth while Rc2 requests a I-Mbps bandwidth.



Router R3, which needs to make a bandwidth reservation, merges the two requests. The reservation is made for 2 Mbps, the larger of the two, because a 2-Mbps input reservation can handle both requests. The same situation is true for R2.

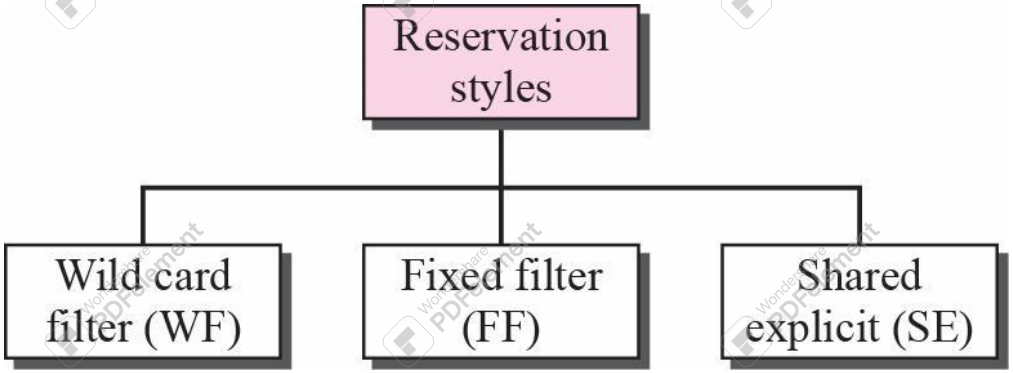
Q-why Rc2 and Rc3, both belonging to one single flow, request different amounts of bandwidth?

Answer: in a multimedia environment, different receivers may handle different grades of quality. For example, Rc2 may be able' to receive video only at 1

Mbps (lower quality), while Rc3 may be able to receive video at 2 Mbps (higher quality).

Reservation Styles

When there is more than one flow, the router needs to make a reservation to accommodate all of them. RSVP defines three types of reservation styles, as shown in Figure.



Wild Card Filter Style In this style, the router creates a single reservation for all senders. The reservation is based on the largest request. This type of style is used when the flows from different senders do not occur at the same time.

Fixed Filter Style In this style, the router creates a distinct reservation for each flow. This means that if there are n flows, n different reservations are made. This type of style is used when there is a high probability that flows from different senders will occur at the same time.

Shared Explicit Style In this style, the router creates a single reservation which can be shared by a set of flows.

Soft State

The reservation information (state) stored in every node for a flow needs to be refreshed periodically. This is referred to as a soft state as compared to the hard state used in other virtual-circuit protocols such as ATM or Frame Relay, where the information about the flow is maintained until it is erased. The default interval for refreshing is currently 30 s.

Problems with Integrated Services

There are at least two problems with Integrated Services that may prevent its full implementation in the Internet: scalability and service-type limitation.

Scalability

The Integrated Services model requires that each router keep information for each flow. As the Internet is growing every day, this is a serious problem.

Service-Type Limitation

The Integrated Services model provides only two types of services, guaranteed and control-load. Those opposing this model argue that applications may need more than these two types of services.

Differentiated Services

Differentiated Services (DS or Diffserv) was introduced by the IETF (Internet Engineering Task Force) to handle the shortcomings of Integrated Services. Two fundamental changes were made:

1. The main processing was moved from the core of the network to the edge of the network. This solves the scalability problem. The routers do not have to store information about flows. The applications, or hosts, define the type of service they need each time they send a packet.
2. The per-flow service is changed to per-class service. The router routes the packet based on the class of service defined in the packet, not the flow. This solves the service-type limitation problem. We can define different types of classes based on the needs of applications.

Q: what are DiffServ goals?

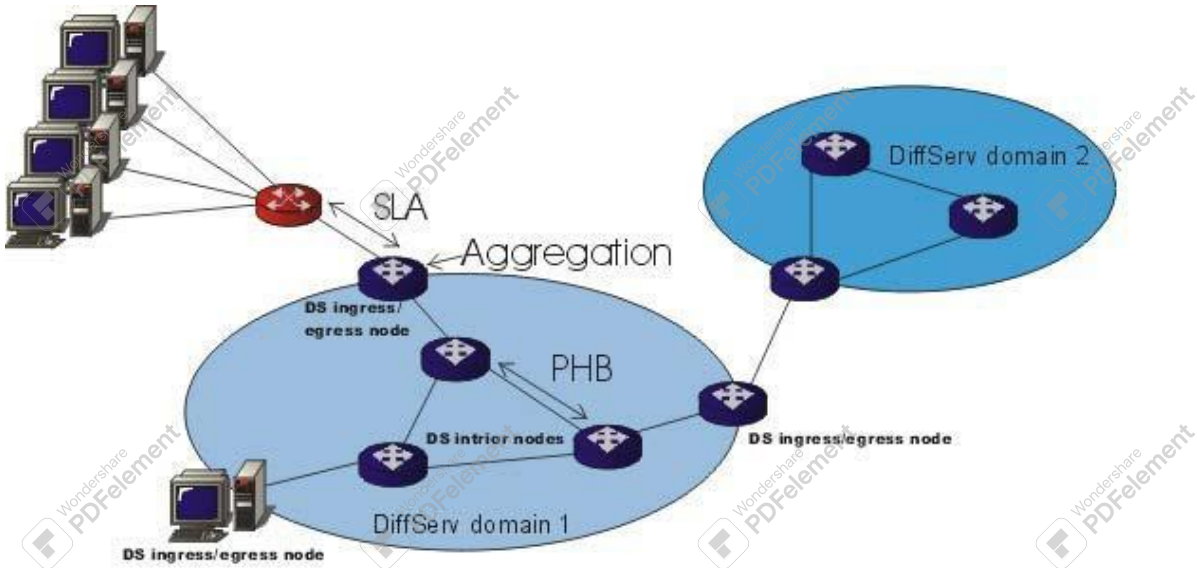
Differentiated Services (DiffServ, DS) is a model to differentiate services on the Internet.

The key concepts are:

- traffic classification and service realization are separated
- each DiffServ domain has its own set of services
- traffic classification is done only at the border of a DiffServ domain
- assume that only a few different static services are required
  - it is sufficient to specify services in long term contracts
  - many flows will receive the same service, i.e. will share the resources of a-service
  - admission and usage control is necessary in order to guarantee a specific QoS.

Example

DiffServ Domains



SLA = Service Level Agreement, between user and provider.
Aggregation = all traffic flows that will receive the same service.
PHB = Per Hop Behavior, is the externally observable forwarding behavior.

DS Field

In Diffserv, each packet contains a field called the DS field. The value of this field is set at the boundary of the network by the host or the first router designated as the boundary router. IETF proposes to replace the existing TOS (type of service) field in IPv4 or the class field in IPv6 by the DS field, as shown in Figure.

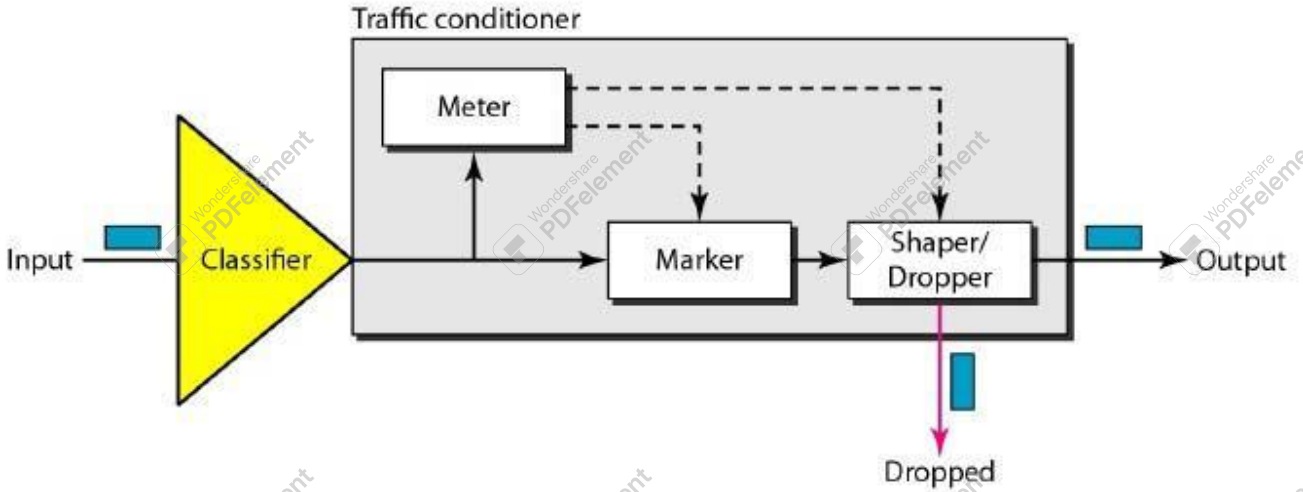
The DS field contains two subfields: DSCP and CU.

The DSCP (Differentiated Services Code Point) is a 6-bit subfield that defines the per-hop behavior (PHB). The 2-bit CU (currently unused) subfield is not

currently used. The Diffserv capable node (router) uses the DSCP 6 bits as an index to a table defining the packet-handling mechanism for the current packet being processed

Traffic Conditioner

To implement Diffserv, the OS node uses traffic conditioners such as meters, markers, shapers, and droppers, as shown in Figure.



Meters The meter checks to see if the incoming flow matches the negotiated traffic profile. The meter also sends this result to other components. The meter can use several tools such as a token bucket to check the profile.

Marker A marker can remark a packet that is using best-effort delivery (OSCP: or down-mark a packet based on information received from the meter. Down marking (lowering the class of the flow) occurs if the flow does not match the profile. A marker does not up-mark (promote the class) a packet.

Shaper A shaper uses the information received from the meter to reshape the traffic if it is not compliant with the negotiated profile.

Dropper A dropper, which works as a shaper with no buffer, discards packets if the flow severely violates the negotiated profile.

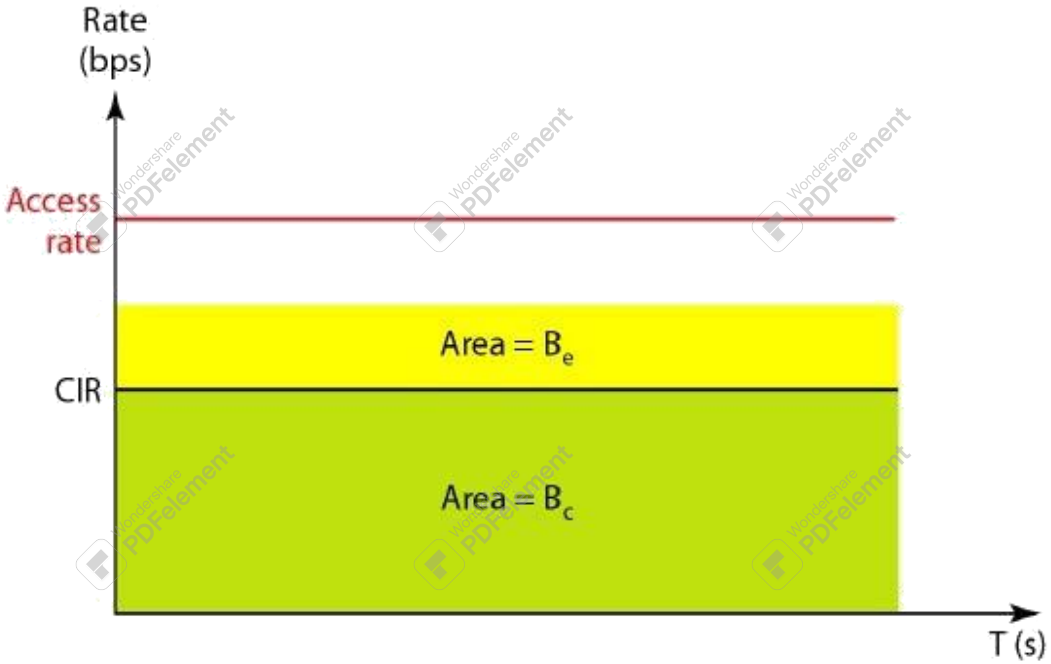
QoS in switched networks

Frame Relay and ATM are two virtual-circuit networks that need a signaling protocol such as RSVP.

QoS in Frame Relay

Four different attributes to control traffic have been devised in Frame Relay: access rate, committed burst size **Be'** committed information rate **(CIR)**, and excess burst size **Be'**.

These are set during the negotiation between the user and the network. For PVC connections, they are negotiated once; for SVC connections, they are negotiated for each connection during connection setup. Figure shows the relationships between these four measurements.



□ Access Rate

For every connection, an access rate (in bits per second) is defined. The access rate actually depends on the bandwidth of the channel connecting the user to the network. The user can never exceed this rate. For example, if the user is connected to a Frame Relay network by a T-l line, the access rate is 1.544 Mbps and can never be exceeded.

□ Committed Burst Size

For every connection, Frame Relay defines a committed burst size **Be**. This is the maximum number of bits in a predefined time that the network is committed to transfer without discarding any frame or setting the DE bit. For example, if a **Be** of 400 kbits for a period of 4 s is granted, the user can send up to 400 kbits during a 4-s interval without worrying about any frame loss. Note that this is not a rate defined for each second. It is a cumulative measurement. The user can send 300 kbits during the first second, no data during the second and the third seconds, and finally 100 kbits during the fourth second.

□ Committed Information Rate

The committed information rate (CIR) is similar in concept to committed burst size except that it defines an average rate in bits per second. If the user follows this rate continuously, the network is committed to deliver the frames. However, because it is an average measurement, a user may send data at a higher rate than the CIR at times or at a lower rate other times. As long as the average for the predefined period is met, the frames will be delivered. The cumulative number

of bits sent during the predefined period cannot exceed **Be** Note that the CIR is not an independent measurement; it can be calculated by using the following formula:

$$\mathbf{CIR} = \frac{B}{T_c} \; bps$$

For example, if the **Be** is 5 kbits in a period of 5 s, the CIR is 5000/5, or 1 kbps.

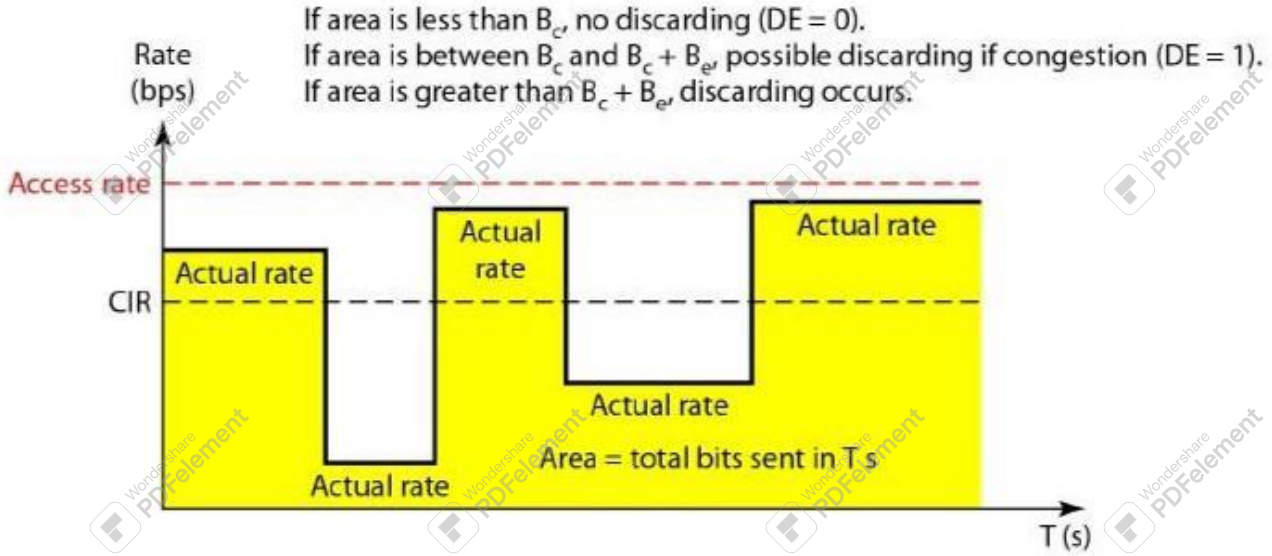
□ Excess Burst Size

For every connection, Frame Relay defines an excess burst size **Be"** This is the maximum number of bits in excess of **Be** that a user can send during a predefined time. The network is committed to transfer these bits if there is no congestion. Note that there is less commitment here than in the case of **Be** The network is committing itself conditionally.

□ User Rate

Figure shows how a user can send bursty data. If the user never exceeds **BC'** the network is committed to transmit the frames without discarding any. If the user

exceeds **Be** by less than **Be** (that is, the total number of bits is less than **Be** + **Be**)' the network is committed to transfer all the frames if there is no congestion. If there is congestion, some frames will be discarded. The first switch that receives the frames from the user has a counter and sets the **DE** bit for the frames that exceed **Be** The rest of the switches will discard these frames if there is congestion. Note that a user who needs to send data faster may exceed the Be level. As long as the level is not above **Be** + **Be'** there is a chance that the frames will reach the destination without being discarded. Remember, however, that the moment the user exceeds the **Be** + **Be** level, all the frames sent after that are discarded by the first switch.

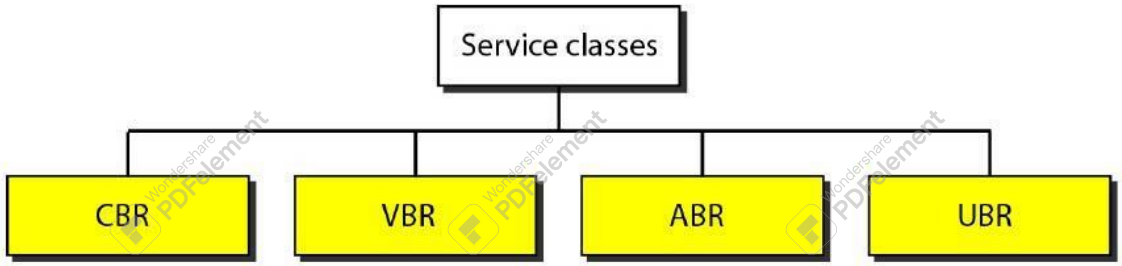


QoS in ATM

The QoS in ATM is based on the class, user-related attributes, and network-related attributes.

Classes

The ATM Forum defines four service classes: CBR, VBR, ABR, and UBR (see Figure.



CBR The constant-bit-rate (CBR) class is designed for customers who need real-time audio or video services. The service is similar to that provided by a dedicated line such as a T line.

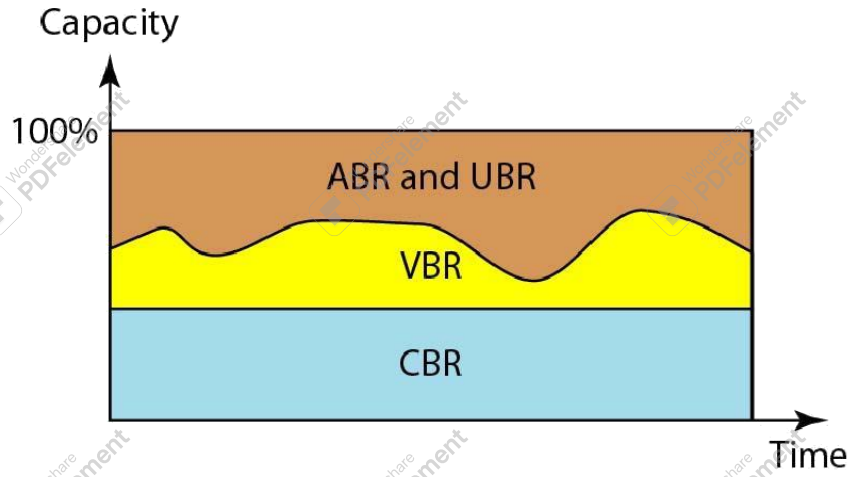
)

VBR The variable-bit-rate (VBR) class is divided into two subclasses: real-time (VBR-RT) and non-real-time (VBR-NRT). VBR-RT is designed for those users who need real-time services (such as voice and video transmission) and use compression techniques to create a variable bit rate. VBR-NRT is designed for those users who do not need real-time services but use compression techniques to create a variable bit rate.

ABR The available-bit -rate (ABR) class delivers cells at a minimum rate. If more network capacity is available, this minimum rate can be exceeded. ABR is particularly suitable for applications that are bursty.

UBR The unspecified-bit-rate (UBR) class is a best-effort delivery service that does not guarantee anything.

Figure shows the relationship of different classes to the total capacity of the network.



User-Related Attributes

ATM defines two sets of attributes. User-related attributes are those attributes that define how fast the user wants to send data. These are negotiated at the time of contract between a user and a network. The following are some user-related attributes:

SCR The sustained cell rate (SCR) is the average cell rate over a long time interval. The actual cell rate may be lower or higher than this value, but the average should be equal to or less than the SCR.

PCR The peak cell rate (PCR) defines the sender's maximum cell rate. The user's cell rate can sometimes reach this peak, as long as the SCR is maintained.

MCR The minimum cell rate (MCR) defines the minimum cell rate acceptable to the sender. For example, if the MCR is 50,000, the network must guarantee that the sender can send at least 50,000 cells per second.

CVDT The cell variation delay tolerance (CVDT) is a measure of the variation in cell transmission times. For example, if the CVDT is 5 ns, this means that the difference between the minimum and the maximum delays in delivering the cells should not exceed 5 ns.

Network-Related Attributes

The network- related attributes are those that define characteristics of the network. The following are some network-related attributes:

CLR The cell loss ratio (CLR) defines the fraction of cells lost (or delivered so late that they are considered lost) during transmission. For example, if the sender sends 100 cells and one of them is lost, the CLR is

$$CLR \frac{1}{100} 10^{2}$$

CTD The cell transfer delay (CTD) is the average time needed for a cell to travel from source to destination. The maximum CTD and the minimum CTD are also considered attributes.

CDV The cell delay variation (CDV) is the difference between the CTD maximum and the CTD minimum.

CER The cell error ratio (CER) defines the fraction of the cells delivered in error.