No.132, AECS Layout
I.T. Park Road, Bangalore 560 037
T: +91 80 28524466/77 F: +91 80 28524630
E: info@cmrit.ac.in | www.cmrit.ac.in

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

University Question Paper Answer Key

December 2017

15CS52 Computer Networks

Module-1

1. a. Compare client-server and peer-to-peer architecture. (5)
    - The key difference between Client-Server and Peer-to-Peer network is that there is a dedicated server and specific clients in the client-server network model whereas, in peer-to-peer each node can act as both server and client.
    - In the client-server model, the server provides services to the client. However, in peer-to-peer, each peer can provide services and can also request for the services.
    - In the client-server model, sharing information is more important whereas, in peer-to-peer model connectivity between peers is more important.
    - In the client-server model, data is stored on a centralized server where as, in peer-to-peer each peer has its own data.
    - In peer-to-peer model, the servers are distributed in a system, so there are fewer chances of server getting bottlenecked, but in the client-server model, there is a single server serving the clients, so there are more chances of server getting bottlenecked.
    - The client-server model is more expensive to implement than peer-to-peer.
    - The client-server model is more scalable and stable than peer-to-peer.

   b. Describe HTTP with persistent and non-persistent connections (8)

The Hypertext Transfer Protocol (HTTP) is an application-level protocol that uses TCP as an underlying transport and typically runs on port 80. HTTP is a stateless protocol i.e. server maintains no information about past client requests.

HTTP Connections

1. Non-Persistent
2. Persistent

Before starting with persistent and non-persistent HTTP connection lets know what is a RTT.

RTT-> Time for a small packet to travel from client to server and back.

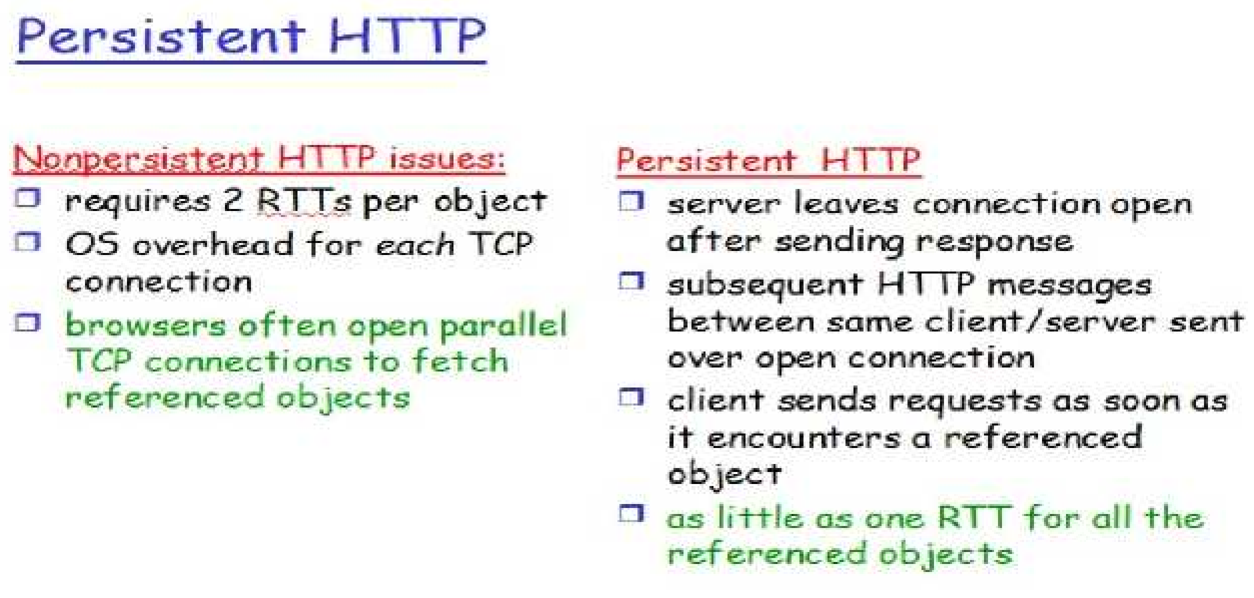
RTT= 2*propagation time

1. For an connection Persistent or Non-persistent it is sure that to initiate TCP connection one RTT is used.
2. One RTT is used for HTTP request and first few bytes to HTTP response to return.

So in order to know total file transmission time->

total = 2RTT+transmit time

Difference between Persistent & Non-Persistent connection.



Persistent HTTP

Nonpersistent HTTP issues:
- requires 2 RTTs per object
- OS overhead for each TCP connection
- browsers often open parallel TCP connections to fetch referenced objects

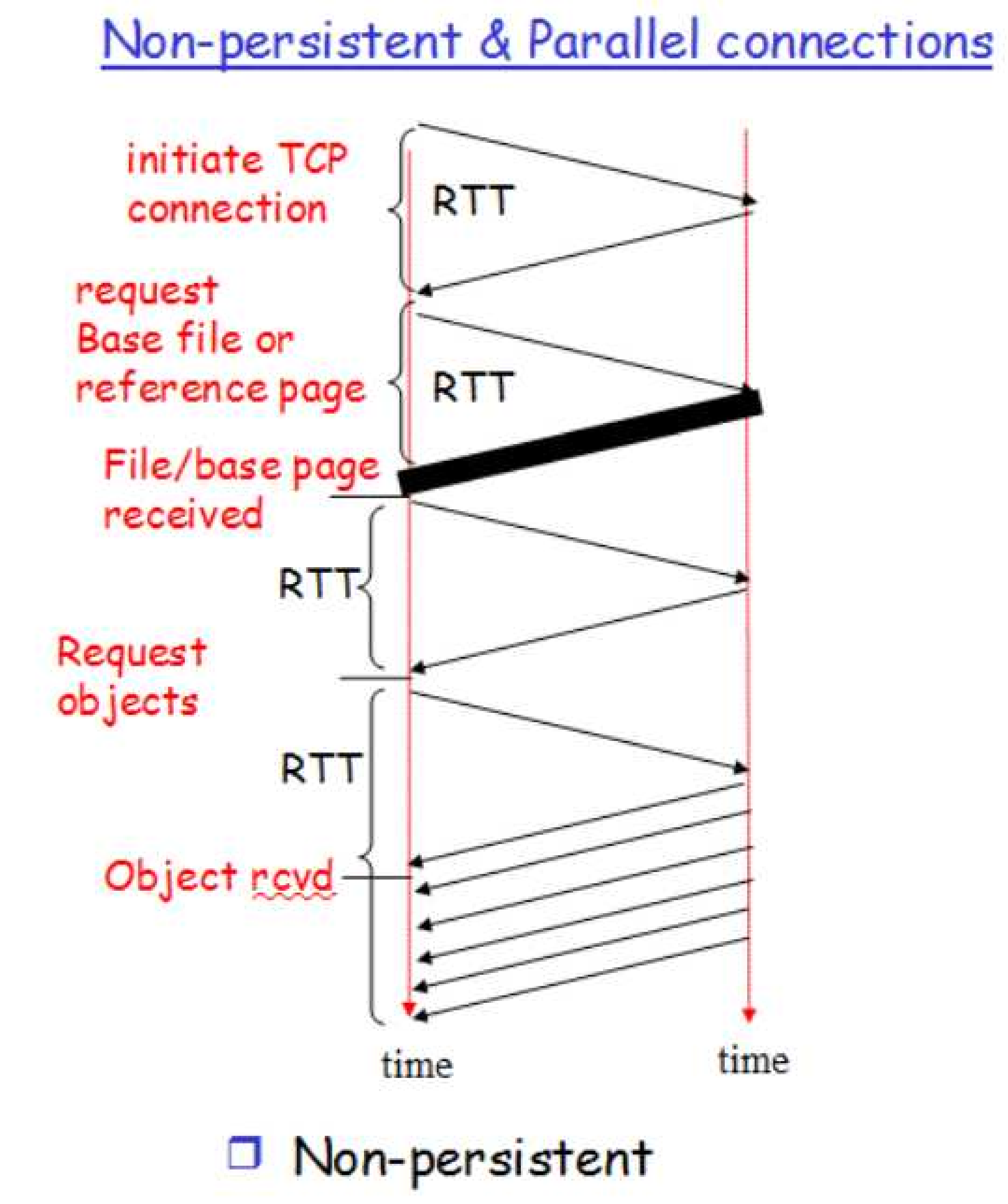
Persistent HTTP
- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over open connection
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects

Non-Persistent Connection

1. Without parallel connection
2. With parallel connection

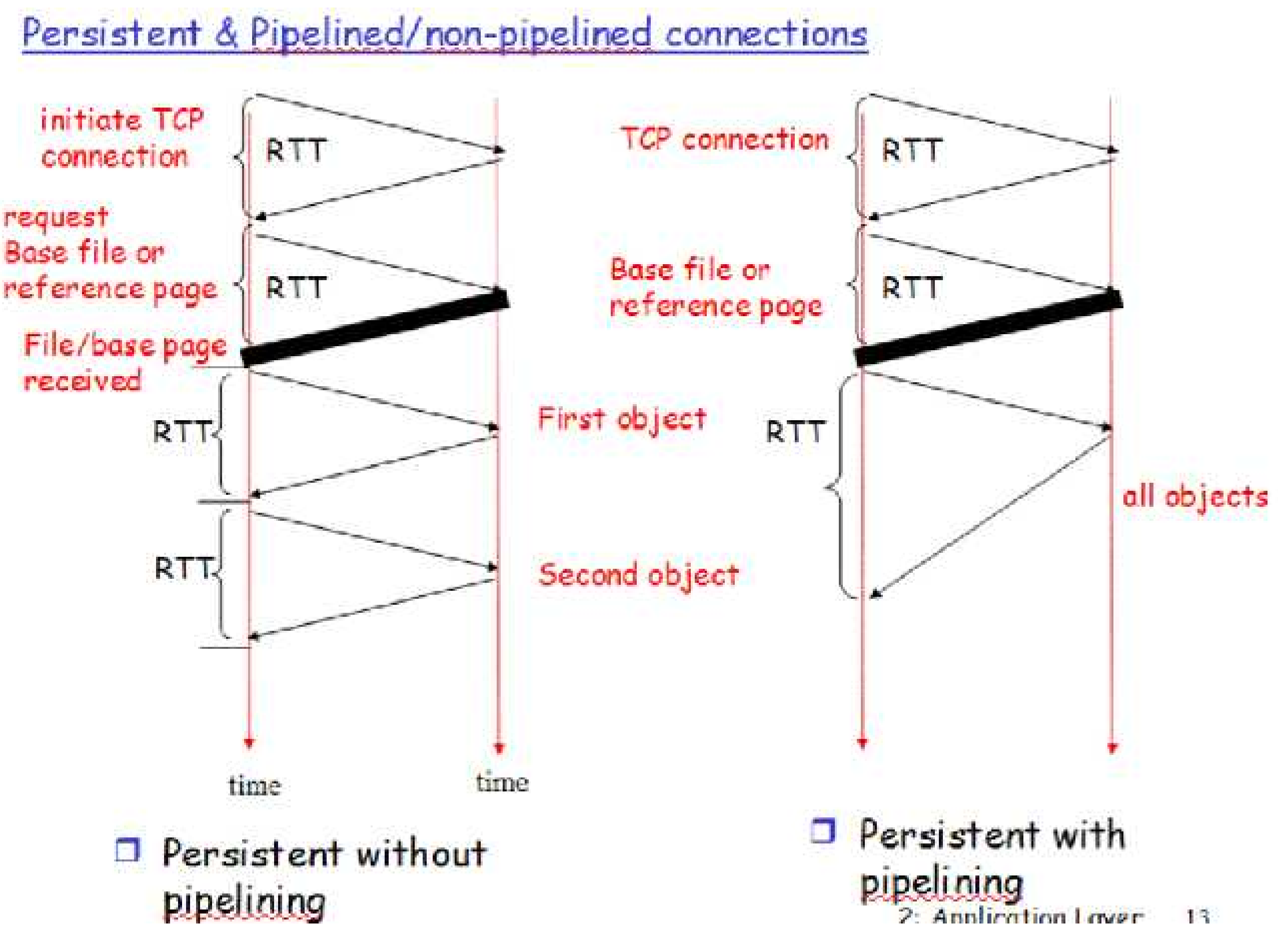
Without parallel connection Non-Persistent
Each objection takes two RTT (assuming no window limit) one for TCP connection and other for HTTP image/text file.

With parallel connection Non-Persistent



Persistent connection

1. Non-Pipelined
2. Pipelined


Persistent & Pipelined/non-pipelined connections
initiate TCP connection
RTT
request Base file or reference page
RTT
File/base page received
RTT
First object
RTT
Second object
time
time
Persistent without pipelining
TCP connection
RTT
Base file or reference page
RTT
RTT
all objects
Persistent with pipelining
2: Application Layer 13

In Non-pipeline connection we first establish connection which takes two RTT then we send all the objects images/text files which takes 1 RTT each (TCP for each object is not required).

In Pipelined connection 2RTT for connection establishment and then 1RTT(assuming no window limit) for all the objects i.e. images/text.

Advantages

1) Lower CPU and memory usage because there are less number of connections.
2) Allows HTTP pipelining of requests and responses.
3) Reduced network congestion (fewer TCP connections).
4) Reduced latency in subsequent requests (no handshaking).
5) Errors can be reported without the penalty of closing the TCP connection.

Disadvantages
Resources may be kept occupied even when not needed and may not be available to others.

Most of the modern browsers like Chrome, Firefox and Internet Explorer use persistent connections.

c. What are the services provided by DNS? (3)

Host Aliasing

Mail Server Aliasing

Load Distribution

2. a. Demonstrate socket implementation using TCP. (8)

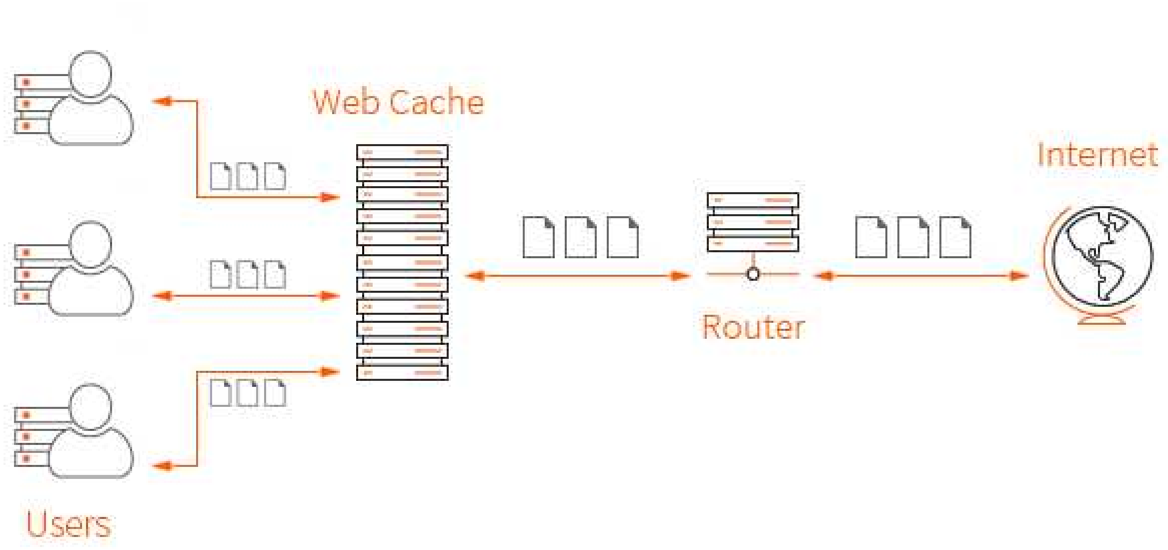
```
packagecom.cn_lab;
importjava.io.BufferedReader;
importjava.io.DataInputStream;
importjava.io.DataOutputStream;
importjava.io.IOException;
importjava.io.InputStreamReader;
importjava.net.Socket;
public class TcpClient {
public static void main(String args[])
{
try {
Socket socket=new Socket("localhost",4555);
// reading the file name from keyboard. Uses inputstream
BufferedReaderbufferedReader=new                    BufferedReader(new
InputStreamReader(System.in));
System.out.println("Enter the file name");
String file_name=bufferedReader.readLine();
DataOutputStreamdout=new DataOutputStream(socket.getOutputStream());
// sending the file name to server.
dout.writeUTF(file_name);
DataInputStream din=new DataInputStream(socket.getInputStream());
String content;
do
{
// receiving the contents from server.  Uses inputstream
content = din.readUTF();
System.out.println(content);
dout.flush();
if(content==null)
{
break;
}
} while(true);
dout.close();
socket.close();
bufferedReader.close();
}
catch(IOException e)
{
}
}}

packagecom.cn_lab;
importjava.io.BufferedReader;
importjava.io.DataInputStream;
importjava.io.DataOutputStream;
importjava.io.FileReader;
importjava.net.ServerSocket;
importjava.net.Socket;
public class TCpServer {
public static void main(String args[]) throws Exception {
// establishing the connection with the server
ServerSocketserverSocket=new ServerSocket(4555);
// binding with port: 4555
Socket socket=serverSocket.accept();
// reading the file name from client
DataInputStream din=new DataInputStream(socket.getInputStream());
String fileName=din.readUTF();
// reading file contents
BufferedReaderbufferedReader=new BufferedReader(new FileReader(fileName));
String content;
// keeping output stream ready to send the contents
DataOutputStreamdout=new DataOutputStream(socket.getOutputStream());
// reading line-by-line from file
while((content=bufferedReader.readLine())!=null&&content.length()!=0){
// sending each line to client
dout.writeUTF(content);
}
//closing resources
```

```
bufferedReader.close();
serverSocket.close();
}
```

b. Write a note on web caching. (4)

A web cache is a hardware device or software application for temporarily storing frequently-accessed static content.



Overview

Internet users with similar interests often download the same web content over and over again. Without a proper web cache, every time a user makes a request the response must come all the way from the origin server (part of the "Internet" in the illustration above). When many users are requesting content at the same time, response times may increase and a server overload may even occur.

A web cache handles requests for popular content that would otherwise be directed to the origin server. It also puts popular content closer to end users, thereby improving response times. One popular type of web cache is a CDN.

How a Web Cache Works

Whenever content is downloaded from the origin server, a copy is stored in the web cache for a set period of time determined by caching rules you set. If another user requests the same content again, the web cache sends the stored content and the user request does not have to be forwarded to origin server again. This is known as content caching.

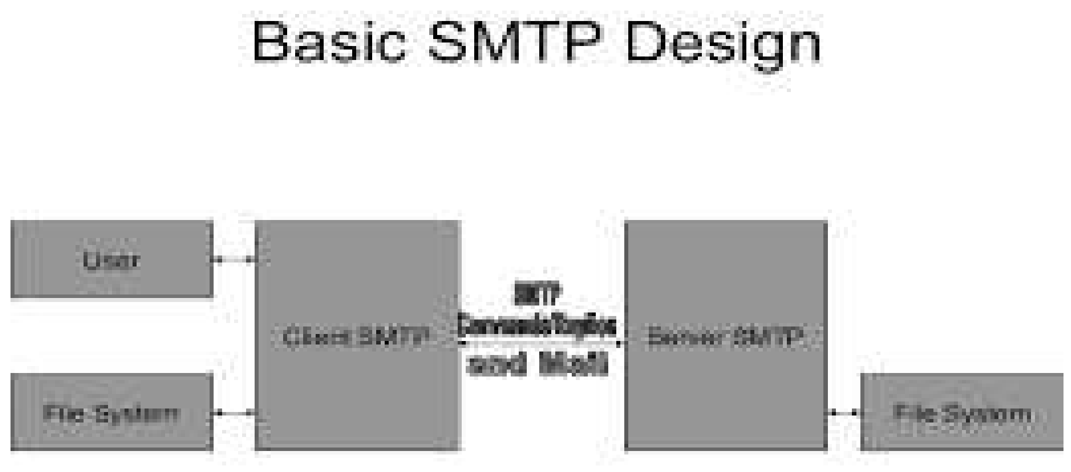
A typical web cache flow:

1. A user accesses a website.
2. The browser sends an HTTP request to the web cache.
3. If the requested object IS stored in the cache, the web cache responds with the object.

   If the requested object IS NOT stored in the cache, the web cache requests the object from the origin server and sends the response to the browser.
4. If the object is cacheable, the web cache retains a copy of the object so that subsequent requests are served locally from the web cache.

c. Illustrate the basic operation of SMTP with an example. (4)

SMTP stands for Simple Mail Transfer Protocol as the name suggest it is basically used for emails.. SMTP specified for outgoing mail uses and default sits on TCP Port 25..(Uses 587 in new versions) Electronic mail servers uses SMTP for send and receive emails while the User Client interface only uses SMTP mainly for sending messages to the mail servers.
For receving emails the User Client applications mainly use POP (Post office Protocol) and IMAP (Internet Messaging Access Protocol).
For receving emails the User Client applications mainly use POP (Post office Protocol) and IMAP (Internet Messaging Access Protocol)..



Module-2

3. a. Elaborate the three way handshaking in TCP. (5)

The process of communication between devices over the internet happens according to the current TCP/IP suite model(stripped out version of OSI reference model). The Application layer is a top pile of stack of TCP/IP model from where network referenced application like web browser on the client side establish connection with the server. From the application layer,the information is transferred to the transport layer where our topic comes into picture. The two important protocols of this layer are – TCP, UDP(User Datagram Protocol) out of which TCP is prevalent(since it provides reliability for the connection established). However you can find application of UDP in querying the DNS server to get the binary equivalent of the Domain Name used for the website.

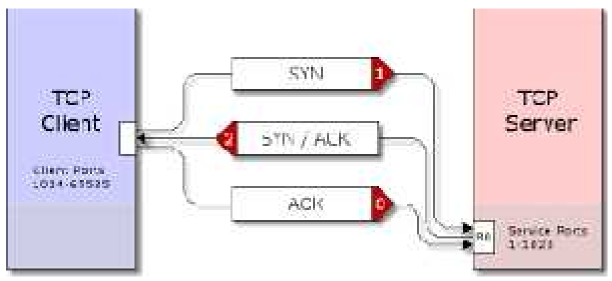


Image Source –TCP 3 way handshake

TCP provides reliable communication with something called Positive Acknowledgement with Re-transmission(PAR). The Protocol Data Unit(PDU) of the transport layer is called segment. Now a device using PAR resend the data unit until it receives an acknowledgement. If the data unit received at the receiver's end is damaged(It checks the data with checksum functionality of the transport layer that is used for Error Detection), then receiver discards the segment. So the sender has to resend the data unit for which positive acknowledgement is not received. You can realize from above mechanism that three segments are exchanged between sender(client) and receiver(server) for a reliable TCP connection to get established. Let us delve how this mechanism works :

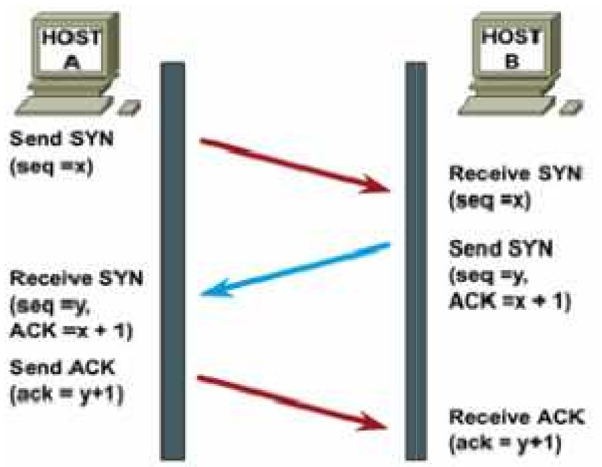


Image Source –Connection Establishment in TCP

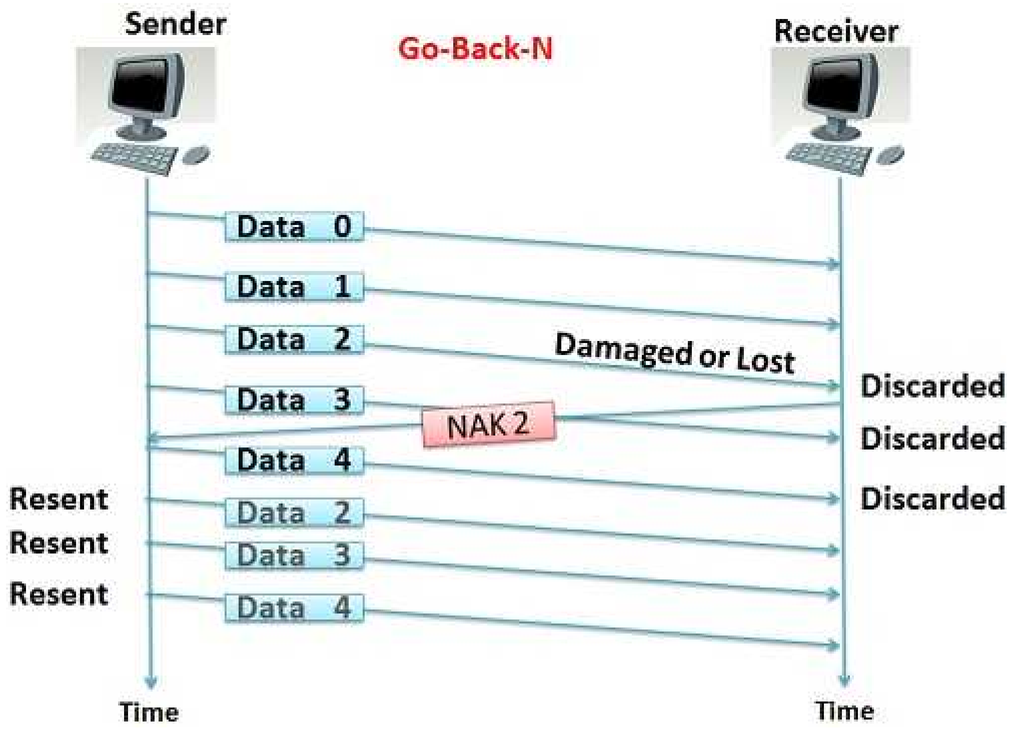
- Step 1 (SYN) : In the first step, client wants to establish a connection with server, so it sends a segment with SYN(Synchronize Sequence Number) which informs server that client is likely to start communication and with what sequence number it starts segments with
- Step 2 (SYN + ACK): Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with
- Step 3 (ACK) : In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start eh actual data transfer

  The steps 1, 2 establish the connection parameter (sequence number) for one direction and it is acknowledged. The steps 2, 3 establish the connection parameter (sequence number) for the other direction and it is acknowledged. With these, a full-duplex communication is established.

b. Discuss Go-Back N protocol. (6)

Definition of Go-Back-N

Go-Back-N protocol is a sliding window protocol. It is a mechanism to detect and control the error in datalink layer. During transmission of frames between sender and receiver, if a frame is damaged, lost, or an acknowledgement is lost then the action performed by sender and receiver is explained in the following content.



Damaged Frame

If a receiver receives a damaged frame or if an error occurs while receiving a frame then, the receiver sends the NAK (negative acknowledgement) for that frame along with that frame number, that it expects to be retransmitted. After sending NAK, the receiver discards all the frames that it receives, after a damaged frame. The receiver does not send any ACK (acknowledgement) for the discarded frames. After the sender receives the NAK for the damaged frame, it retransmits all the frames onwards the frame number referred by NAK.

Lost frame

The receiver checks the number on each frame, it receives. If a frame number is skipped in a sequence, then the receiver easily detects the loss of a frame as the newly received frame is received out of sequence. The receiver sends the NAK for the lost frame and then the receiver discards all the frames received after a lost frame. The receiver does not send any ACK (acknowledgement) for that discarded frames. After the sender receives the NAK for the lost frame, it retransmits the lost frame referred by NAK and also retransmits all the frames which it has sent after the lost frame.

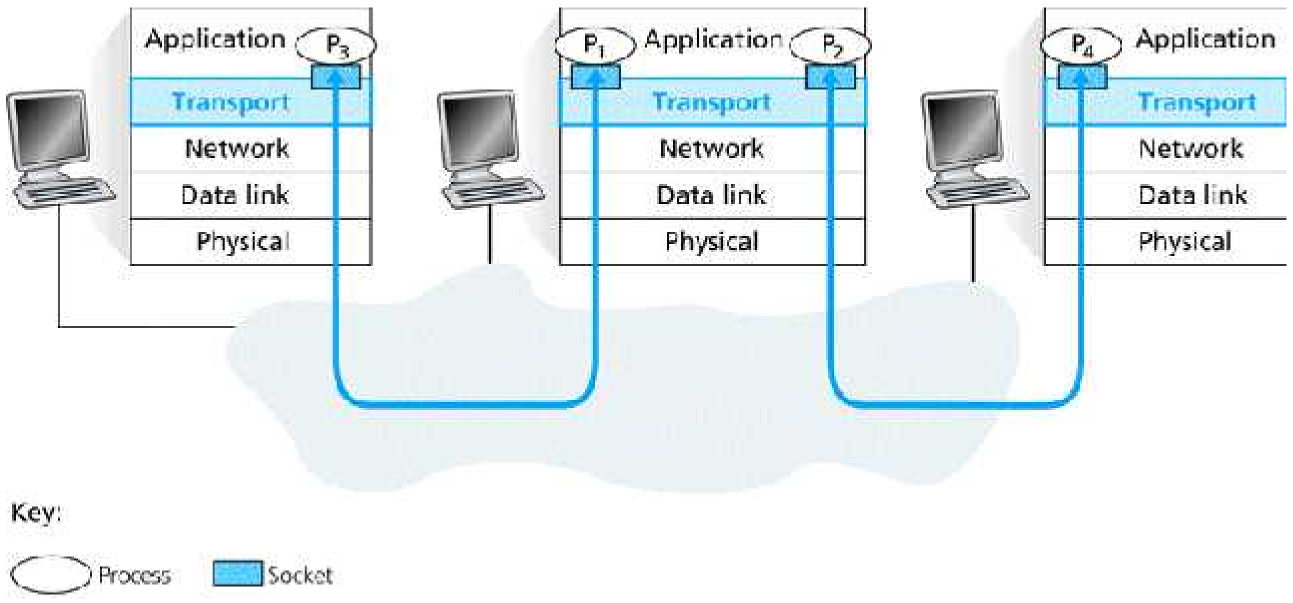
Lost Acknowledgement

If the sender does not receive any ACK or if the ACK is lost or damaged in between the transmission. The sender waits for the time to run out and as the time run outs, the sender retransmits all the frames for which it has not received the ACK. The sender identifies the loss of ACK with the help of a timer.

The ACK number, like NAK (negative acknowledgement) number, shows the number of the frame, that receiver expects to be the next in sequence. The window size of the receiver is 1 as the data link layer only require the frame which it has to send next to the network layer. The sender window size is equal to 'w'. If the error rate is high, a lot of bandwidth is lost wasted.

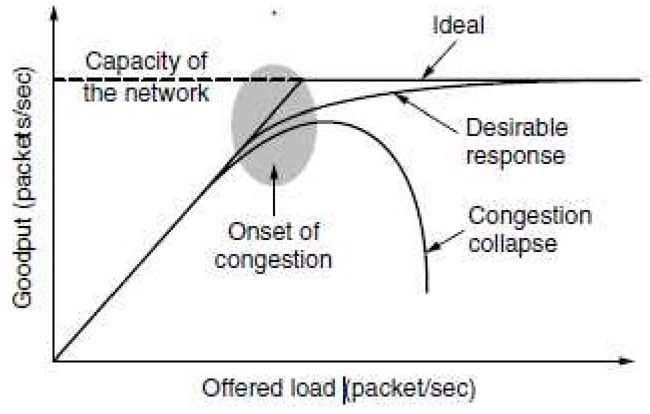
c. Explain the connection-oriented multiplexing and de-multiplexing. (5)

- A socket is the interface through which a process (application) communicates with the transport layer
- each process can potentially use many sockets
- the transport layer in a receiving machine receives a sequence of segments from its network layer
- delivering segments to the correct socket is called demultiplexing
- assembling segments with the necessary information and passing them to the network layer is called multiplexing
- multiplexing and demultiplexing are need whenever a communications channel is shared



4. a. State congestion and discuss the cause of congestion. (4)

Too many packets present in (a part of) the network causes packet delay and loss that degrades performance. This situation is called congestion. The network and transport layers share the responsibility for handling congestion.



With too much traffic the performance drops sharply

Common reasons for congestion on router - Congestion can occur on a router when packets arrive at a greater rate than possible to forward. Congestion can be sporadic or long term. When congestion occurs, packets must be discarded by the router. Congestion occurs at a bottleneck when:

i. Packets arrive on several channels to be forwarded on a single channel.

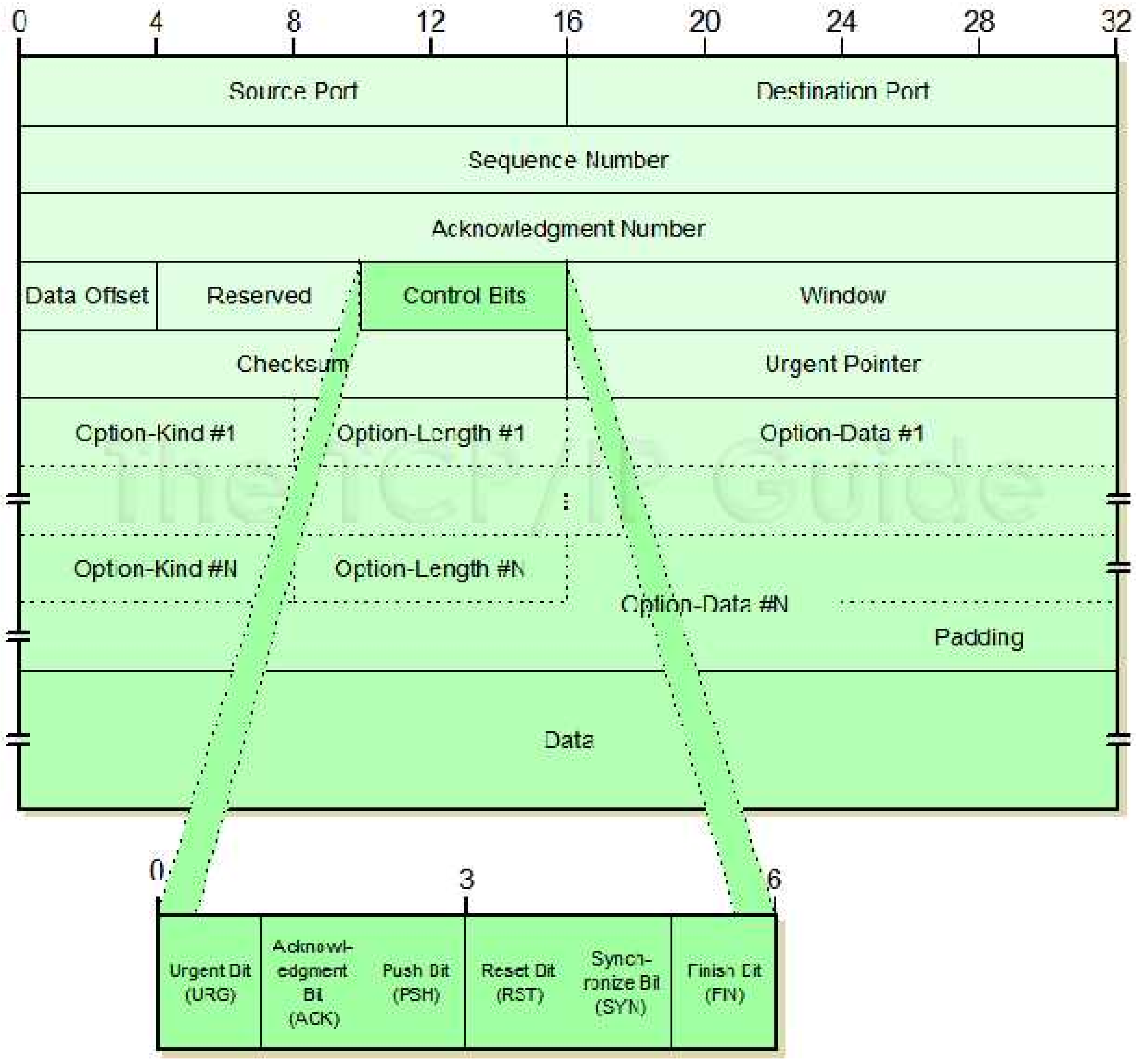
ii. Incoming channel has a higher bandwidth than outgoing channel.

iii. Channel bandwidth is sufficient but router CPU processing is too slow to handle bookkeeping (queuing, routing table updates, etc).

iv. Router lacks sufficient memory buffer space.

v. Under an end-to-end reliable protocol (e.g. TCP), even with infinite memory, congestion can get worse because packets have timed out (e.g. moving from queue back to front on router or due to delay in acknowledging on a receiver) resulting in duplicates, adding to the congestion.

b. With a neat-diagram explain the TCP segment structure. (8)



c. Suppose that two measured sample RTT values are 106ms and 120 ms. Compute:

i) Estimated RTT after each of these sample RTT value is obtained. Assume α=0.125 and estimated RTT is 100msec just before first of the samples obtained.

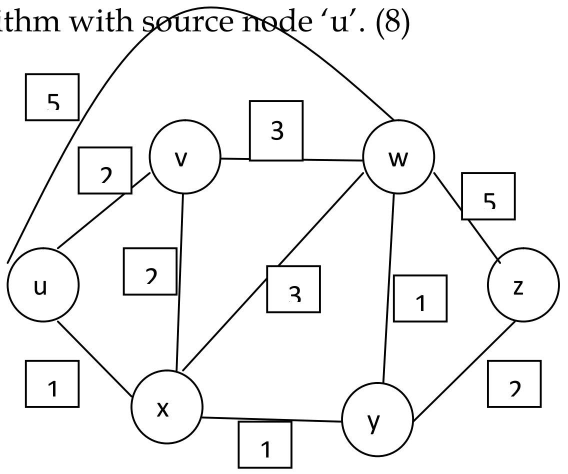
EstimatedRTT = (1-α)*EstimatedRTT_prev + α*SampleRTT
=(1-0.125)*100+0.125*113
=87.5+14.125= 102ms

ii) Compute DevRTT.
Assume β=0.25 and DevRTT was 5 msec before first of these samples are obtained. (4)

DevRTT = (1-β)*DevRTT_prev + β*|SampleRTT - EstimatedRTT|
=(1-0.25)*5+0.25*(113-102)
=3.75+2.75=7ms

Module-3

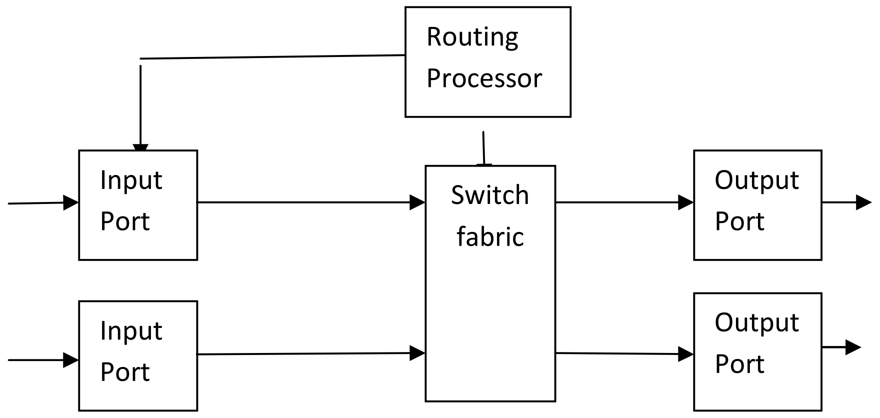
5. a. Write the link state routing algorithm. Solve the following graph using link-state algorithm with source node 'u'. (8)



Step	N'	V	W	X	Y	Z
0	u	2,u	5,u	1,u	α	α
1	ux	2,u	4,x		2,x	α
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

b. What is routing? Explain the structure of a router. (8)
Routing:
The process of finding the optimal path to the destination from the source.

Router architecture



Components:

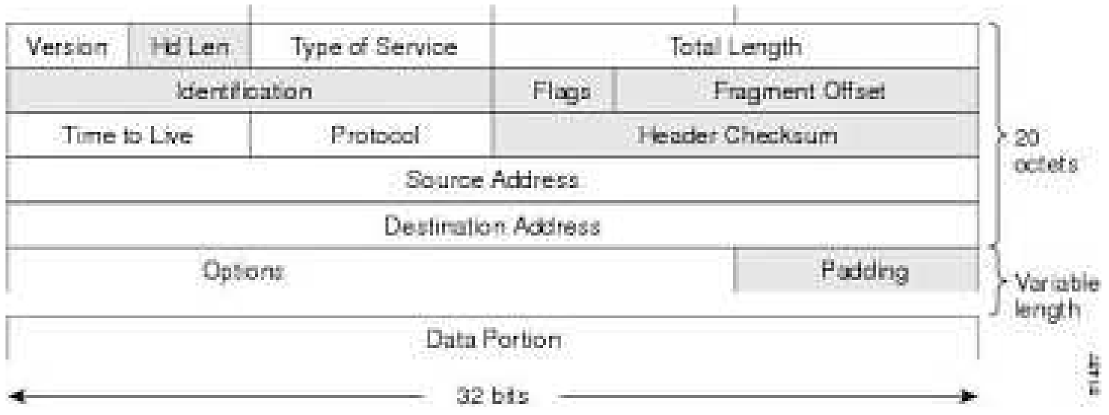
- Input ports. An input port performs several key functions. It performs the physical layer function of terminating an incoming physical link at a router; this is shown in the leftmost box of the input port and the rightmost box of the output port in Figure. An input port also performs link-layer functions needed to interoperate with the link layer at the other side of the incoming link; this is represented by the middle boxes in the input and output ports. Perhaps most crucially, the lookup function is also performed at the input port; this will occur in the rightmost box of the input port. It is here that the forwarding table is consulted to determine the router output port to which an arriving packet will be forwarded via the switching fabric. Control packets (for example, packets carrying routing protocol information) are forwarded from an input port to the routing processor.
- Switching fabric: The switching fabric connects the router's input ports to its output ports. This switching fabric is completely contained within the router— a network inside of a network router.
- Output ports: Output port stores packets received from the switching fabric and transmit these packets on the outgoing link by performing the necessary link-layer and physical-layer functions. When a link is bidirectional (that is, carries traffic in

both directions), an output port will typically be paired with the input port for that link on the same line card (a printed circuit board containing one or more input ports, which is connected to the switching fabric).

- Routing processor: The routing processor executes the routing protocols maintains routing tables and attached link state information, and computes the forwarding table for the router.

A router's input ports, output ports, and switching fabric together implement the forwarding function and are almost always implemented in hardware. These forwarding functions are sometimes collectively referred to as the router forwarding plane. Forwarding plane hardware can be implemented either using a router vendor's own hardware designs, or constructed using purchased merchant-silicon chips (e.g., as sold by companies such as Intel and Broadcom). While the forwarding plane operates at the nanosecond time scale, a router's control functions—executing the routing protocols, responding to attached links.

6. a. Discuss the IPv6 packet format. (5)



b. Elaborate the path attributes in BGP and steps to select the BGP routes. (5)

Service providers working with IP networks are very clear that the Border Gateway Protocol, or BGP, is the most complex and difficult-to-configure internet protocol. Its emphasis on security and scalability makes it essential, however. This tutorial gives you a detailed look at how BGP works, and offers simple and advanced BGP troubleshooting options, so your BGP-enabled routers can exchange information securely with several hundred thousand IP prefixes and keep the internet running.

If you have to explain what BGP is to someone new to the service provider environment, the best definition is that it's the routing protocol that makes the internet work. As the address allocation in the internet is not nearly as hierarchical as the telephone dialing plan, most of the routers in service providers' core networks need to exchange information about several hundred thousand IP prefixes. BGP is still able to accomplish that task, which is proof it's a highly scalable routing protocol.

BGP routing information is usually exchanged between competing business entities in the form of internet service providers (ISPs) in an open, hostile environment -- the public internet. BGP is very security-focused -- for example, all adjacent routers have to be configured manually -- and decent BGP implementations provide a rich set of route filters to allow ISPs to defend their networks and control what they advertise to their competitors.

How BGP works

In BGP terminology, an independent routing domain, which almost always means an ISP network, is called an autonomous system.

BGP is always used as the routing protocol of choice between ISPs (known as external BGP), but also as the core routing protocol within large ISP networks (known as internal BGP).

All other routing protocols are concerned solely with finding the optimal path toward all known destinations. BGP cannot take this simplistic approach because the peering agreements between ISPs almost always result in complex routing policies. To help network operators implement these policies, BGP carries a large number of attributes with each IP prefix, for example:

- Autonomous system (AS) path -- the complete path documenting which autonomous systems a packet would have to travel through to reach the destination.
- Local preference -- the internal cost of a destination, which is used to ensure AS-wide consistency.
- Multi-exit discriminator -- this attribute gives adjacent ISPs the ability to prefer one peering point over another.
- Communities -- a set of generic tags that can be used to signal various administrative policies between BGP routers.

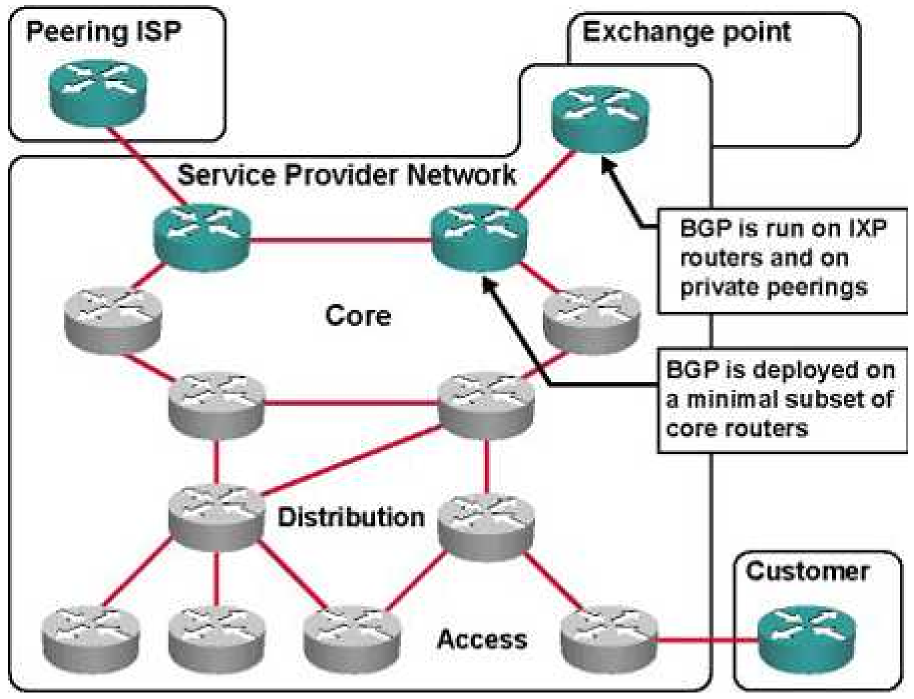
Because the focus of BGP design and implementation was always on security and scalability, it is harder to configure than other routing protocols, more complex -- more so when you start configuring various routing policies -- and one of the slowest converging routing protocols.

The slow BGP convergence dictates a two-protocol design of an ISP network:

- An internal routing protocol -- most often, Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS) -- is used to achieve fast convergence for internal routes, including IP addresses of BGP routers.
- BGP is used to exchange internet routes.

A failure within the core network would thus be quickly bypassed, thanks to fast convergence of OSPF or IS-IS, whereas BGP on top of an internal routing protocol would meet the scalability, security and policy requirements. Even more, if you migrate all your customer routes into BGP, the customer problems -- for example, link flaps between your router and customer's router -- will not affect the stability of your core network.

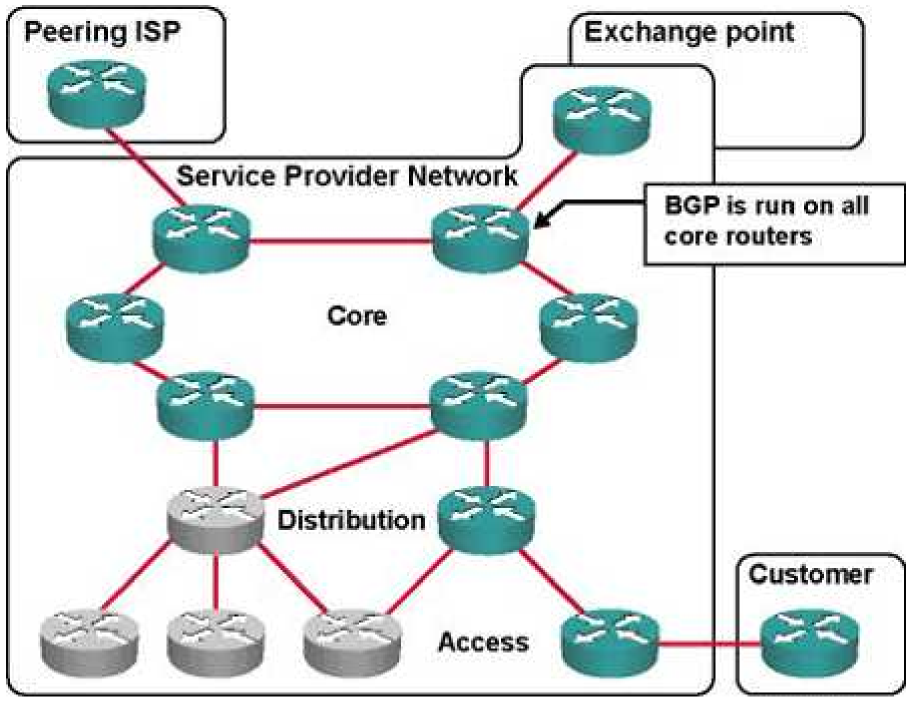
Because of BGP's inherent complexity, customers and small ISPs would deploy BGP only where needed -- for example, on peering points and on a minimal subset of core routers (the ones between the peering points), as shown in the following diagram.


Peering ISP
Exchange point
Service Provider Network
BGP is run on IXP routers and on private peerings
Core
BGP is deployed on a minimal subset of core routers
Distribution
Customer
Access

Minimal BGP deployment

The BGP-speaking routers would also have to generate a default route into the internal routing protocol to attract the traffic for internet destinations not known to other routers in your network.

As your ISP business grows, however, your customers will start requiring BGP connectivity -- any customer who wants to achieve truly redundant internet access has to have its own autonomous system and exchange BGP information with its ISPs -- and you'll be forced to deploy BGP on more and more core and edge routers (see the following diagram). It is, therefore, best you include BGP on all core and major edge routers as part of your initial network design. Even though you might not deploy it everywhere with the initial network deployment, having a good blueprint will definitely help you when you have to scale the BGP-speaking part of your network.



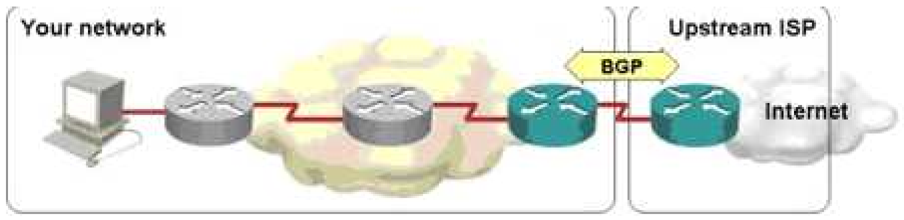
BGP included in network design

BGP requires a full mesh of internal BGP sessions -- sessions between routers in the same autonomous system. You could use BGP route reflectors or BGP confederations to make your network scalable.

Another excellent reason why you'd want to deploy BGP throughout your network is because MPLS-based virtual private networks, large-scale quality-of-service deployments or large-scale differentiated web caching implementations rely on BGP to transport the information they need.

BGP is, without doubt, the most complex IP routing protocol currently deployed in the internet. Its complexity is primarily due to its focus on security and routing policies -- BGP is used to exchange cooperative information (internet routes) between otherwise competing entities (service providers), and it has to be able to implement whatever has been agreed upon in interprovider peering agreements. These agreements often have little to do with technically optimum services.

A structured approach to BGP troubleshooting, however, as illustrated in this and the next section, can quickly lead you from initial problem diagnosis to the solution. Here, we focus on a simple scenario with a single BGP-speaking router in your network (see the following diagram). Similar designs are commonly used by multihomed customers and small internet service providers that do not offer BGP connectivity to their customers.



c. List the broadcast routing algorithms. Explain any one of them. (6)

Broadcast Routing Algorithms

Perhaps the most straightforward way to accomplish broadcast communication is for the sending node to send a separate copy of the packet to each destination, as illustrated in Figure 1(a). Given N destination nodes, the source node simply makes N copies of the packet, addresses each copy to a different destination, and then transmits the N copies to the N destinations using unicast routing. This N-way-unicast approach to broadcasting is simple - no new network-layer routing protocol, packet-duplication, or forwarding functionality is required. There are, however, several drawbacks to this approach. The first drawback is its inefficiency. If the source node is connected to the rest of the network via a single link, then N separate copies of the (same) packet will traverse this single link. It would clearly be more efficient to send only a single copy of a packet over this first hop and then have the node at the other end of the first hop make and forward any additional needed copies. That is, it would be more efficient for the network nodes themselves (rather than just the source node) to create duplicate copies of a packet. For instance, in Figure 1(b), only a single copy of a packet traverses the R1-R2 link. That packet is then duplicated at R2, with a single copy being sent over linksR2-R3andR2-R4.

The additional disadvantages of N-way-unicast are perhaps more subtle, but no less important. An implicit assumption of N-way-unicast is that broadcast recipients, and their addresses, are known to the sender. But how is this information obtained? Most likely, additional protocol mechanisms (such as a broadcast membership or destination-registration protocol) would be needed. This would add more overhead and, importantly, additional complexity to a protocol that had initially seemed quite simple. A final disadvantage of N-way-unicast relates to the purposes for which broadcast is to be used. In "Routing Algorithms", we studied that link-state routing protocols use broadcast to disseminate the link-state information that is used to compute unicast routes. Clearly, in situations where broadcast is used to create and update unicast routes, it would be unwise (at best!) to rely on the unicast routing infrastructure to achieve broadcast.

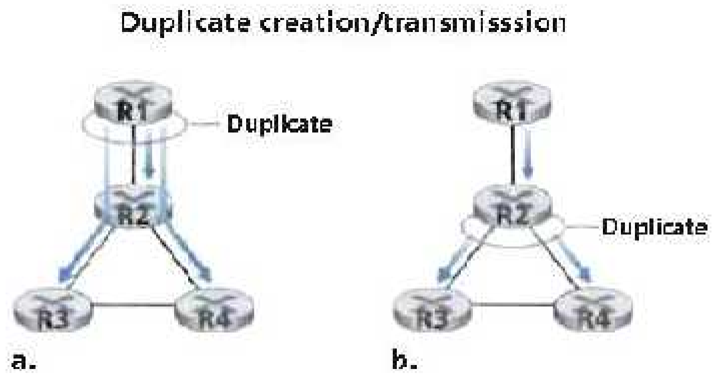


Figure 1. Source-duplication versus in-network duplication

Given the various disadvantages of N-way-unicast broadcast, approaches in which the network nodes themselves play an active role in packet duplication, packet forwarding, and computation of the broadcast routes are clearly of interest. We'll look at several such approaches below and again adopt the graph notation introduced in "Routing Algorithms". We again model the network as a graph, G = (N,E), where N is a set of nodes and a collection E of edges, where each edge is a pair of nodes from N. We'll be a bit sloppy with our notation and use N to refer to both the set of nodes, as well as the cardinality (|N|) or size of that set when there is no confusion.

Uncontrolled Flooding

The most obvious technique for achieving broadcast is a flooding approach in which the source node sends a copy of the packet to all of its neighbors. When a node receives a broadcast packet, it duplicates the packet and forwards it to all of its neighbors (except the neighbor from which it received the packet). Clearly, if the graph is connected, this scheme will eventually deliver a copy of the broadcast packet to all nodes in the graph. Though this scheme is simple and elegant, it has a fatal flaw (before you read on, see if you can figure out this fatal flaw): If the graph has cycles, then one or more copies of each broadcast packet will cycle indefinitely. For example, in Figure 1, R2 will flood to R3, R3 will flood to R4, R4 will flood to R2, and R2 will flood (again!) to R3, and so on. This simple scenario results in the endless cycling of two broadcast packets, one clockwise, and one counterclockwise. But there can be an even more calamitous fatal flaw: When a node is connected to more than two other nodes, it will create and forward multiple copies of the broadcast packet, each of which will create multiple copies of itself (at other nodes with more than two neighbors), and so on. This broadcast storm, resulting from the endless multiplication of broadcast packets, would eventually result in so many broadcast packets being created that the network would be rendered useless.

Controlled Flooding

The key to avoiding a broadcast storm is for a node to judiciously choose when lo flood a packet and (e.g., if it has already received and flooded an earlier copy of a packet) when not to flood a packet. In practice, this can be done in one of many ways.

In sequence-number-controlled flooding, a source node puts its address (or other unique identifier) as well as a broadcast sequence number into a broadcast packet, then sends the packet to all of its neighbors. Each node maintains a list of the source address and sequence number of each broadcast packet it has already received, duplicated, and forwarded. When a node receives a broadcast packet, it first checks whether the packet is in this list. If so, the packet is dropped; if not, the packet is duplicated and forwarded to all the node's neighbors (except the node from which the packet has just been received). The Gnutella protocol, discussed in "Application Layer", uses sequence-number-controlled flooding to broadcast queries in its overlay network. (In Gnutella, message duplication and forwarding is performed at the application layer rather than at thenetworklayer.)

A second approach to controlled flooding is known as reverse path forwarding (RPF) [Dalal 1978], also sometimes referred to as reverse path broadcast (RPB). The idea behind RPF is simple, yet elegant. When a router receives a broadcast packet with a given source address, it transmits the packet on all of its outgoing links (except the one on which it was received) only if the packet arrived on the link that is on its own shortest unicast path back to the source. Otherwise, the router simply discards the incoming packet without forwarding it on any of its outgoing links. Such a packet can be dropped because the router knows it either will receive or has already received a copy of this packet on the link that is on its own shortest path back to the sender. (You might want to convince yourself that this will, actually, happen and that looping and broadcast storms will not occur.) Note that RPF does not use unicast routing to actually deliver a packet to a destination, nor does it require that a router know the complete shortest path from itself to the source. RPF need only know the next neighbor on its unicast shortest path to the sender, it uses this neighbor's identity only to determine whether or not to flood a received broadcast packet.

Figure 2 shows RPF. Suppose that the links drawn with thick lines represent the least-cost paths from the receivers to the source (A). Node A initially broadcasts a source-A packet to nodes C and B. Node B will forward the source-A packet it has received from A (since A is on its least-cost path to A) to both C and D. B will ignore (drop, without forwarding) any source-A packets it receives from any other

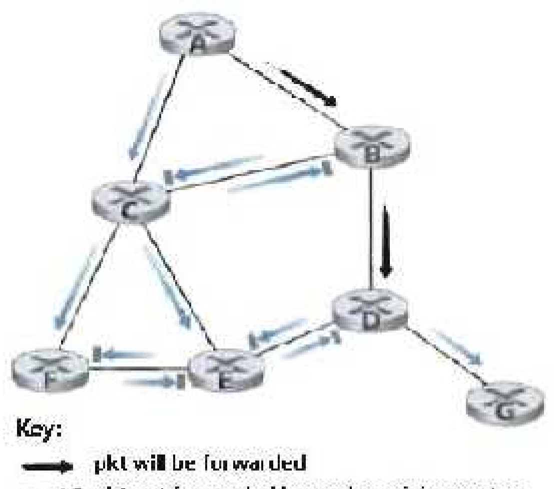


Figure 2. Reverse path forwarding

nodes (for instance, from routers C or D). Let us now examine node C, which will receive a source-A packet directly from A as well as from B. Since B is not on C's own shortest path back to A, C will ignore any source-A packets it receives from B. However, when C receives a source-A packet directly from A, it will forward the packet to nodes B, E, and F.

Spanning-Tree Broadcast

While sequence-number-controlled flooding and RPF avoid broadcast storms, they do not completely avoid the transmission of redundant broadcast packets. For instance, in Figure 3, nodes B, C, D, E, and F receive either one or two redundant packets. Ideally, every node should receive only one copy of the broadcast packet. Examining the tree consisting of the nodes connected by thick lines in Figure 3(a), you can see that if broadcast packets were forwarded only along links within this tree, each and every network node would receive exactly one copy of the broadcast packet - exactly the solution we were looking for! This tree is an example of a spanning tree - a tree that contains each and every node in a graph. More formally, a spanning tree of a graph G = (N,E) is a graph G' = (N,E') such that E' is a subset of E, G' is connected, G' contains no cycles, and G' contains all the original nodes in G. If each link has an associated cost and the cost of a tree is the sum of the link costs, then a spanning tree whose cost is the minimum of all of the graph's spanning trees is called (not surprisingly) a minimum spanning tree.

Therefore, another approach to providing broadcast is for the network nodes to first construct a spanning tree. When a source node wants to send a broadcast packet, it sends the packet out on all of the incident links that belong to the spanning tree. A node receiving a broadcast packet then forwards the packet to all its neighbors in the

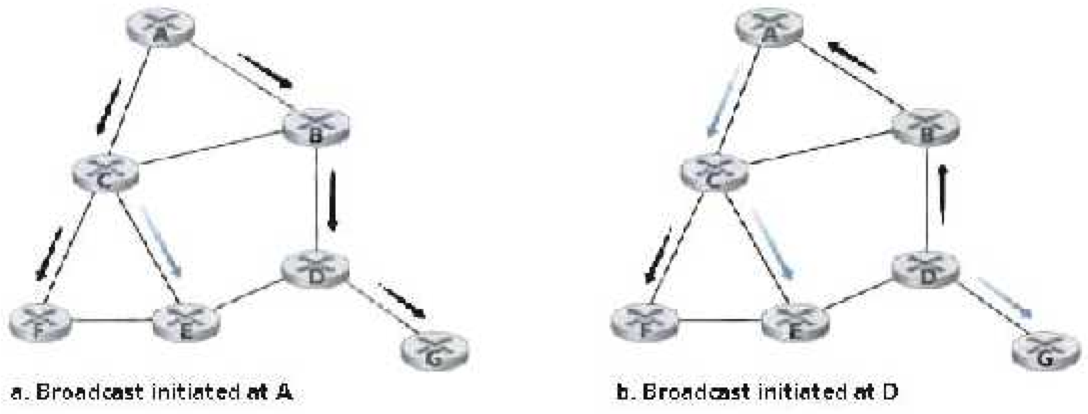


Figure 3. Broadcast along a spanning tree

spanning tree (except the neighbor from which it received the packet). Not only does spanning tree eliminate redundant broadcast packets, but once in place, the spanning tree can be used by any node to begin a broadcast, as illustrated in Figures 3(a) and 3(b). Note that a node need not be aware of the entire tree. The main difficulty associated with the spanning-tree approach is the creation and maintenance of the spanning tree. Several distributed spanning-tree algorithms have been developed [Gallager 1983, Gartner 2003]. We examine only one simple algorithm here. In the center-based approach to building a spanning tree, a center node (also known as a rendezvous point or a core) is defined. Nodes then unicast tree-join messages addressed to the center node. A tree-join message is forwarded using unicast routing toward the center until it either arrives at a node that already belongs to the spanning tree or arrives at the center. In either case, the path that the tree-join message has followed defines the branch of the spanning tree between the edge node that initiated the tree-join message and the center. One can think of this new path as being grafted onto the existing spanning tree.

Figure 4 shows the construction of a center-based spanning tree. Assume that node E is selected as the center of the tree. Assume that node F first joins the tree and forwards a tree-join message to E. The single link EF becomes the initial spanning tree. Node B then joins the spanning tree by sending its tree-join message to E. Assume that the unicast path route to E from B is via D. In this case, the tree-join message results in the path BDE being grafted onto the spanning tree. Node A next joins the spanning group by forwarding its tree-join message towards E. If A's unicast path to E is through B, then since B has already joined the spanning tree, the arrival of A's tree-join message at B will result in the AB link being immediately) grafted onto the spanning tree. Node C joins the spanning tree next by forwarding

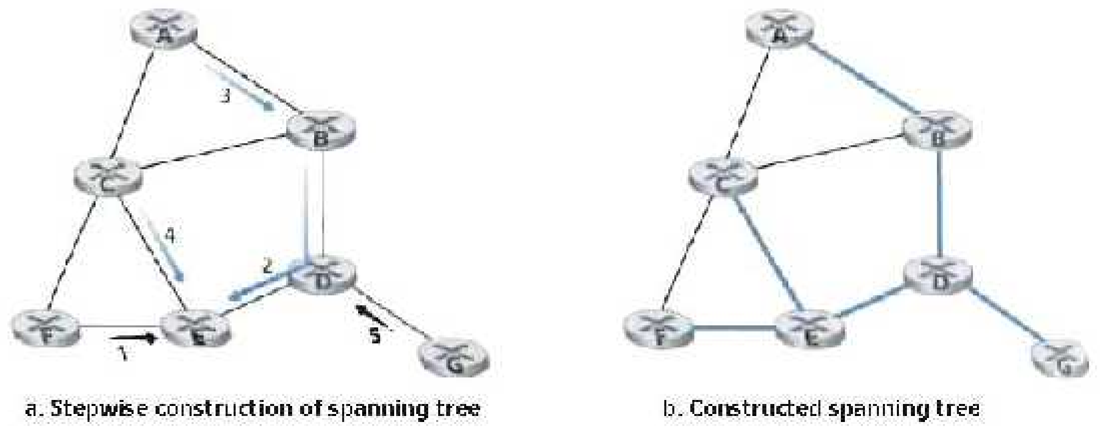

A
B
C
D
E
F
G
1
2
3
4
5
a. Stepwise construction of spanning tree
b. Constructed spanning tree

Figure 4. Center-based construction of a spanning tree

its tree-join message directly to E. Finally, because the unicast routing from G to E must be via node D, when G sends its tree-join message to E, the GD link is grafted onto the spanning tree at node D.

Broadcast Algorithms in Practice

Broadcast protocols are used in practice at both the application and network layers. Gnutella [Gnutella 2009] uses application-level broadcast in order to broadcast queries for content among Gnutella peers. Here, a link between two distributed application-level peer processes in the Gnutella network is actually a TCP connection. Gnutella uses a form of sequence-number-controlled flooding in which a 16-bit identifier and a 16-bit payload descriptor (which identifies the Gnutella message type) are used to detect whether a received broadcast query has been previously received, duplicated, and forwarded. Gnutella also uses a time-to-live (TTL) field to limit the number of hops over which a flooded query will be forwarded. When a Gnutella process receives and duplicates a query, it decrements the TTL field before forwarding the query. Thus, a flooded Gnutella query will only reach peers that are within a given number (the initial value of TTL) of application-level hops from the query initiator. Gnutella's flooding mechanism is thus sometimes referred to as limited-scope flooding.

A form of sequence-number-controlled flooding is also used to broadcast link-state advertisements (LSAs) in the OSPF [Perlman 1999] routing algorithm, and in the Intermediate-System-to-Intermediate-System (IS-IS) routing algorithm [Perlman 1999]. OSPF uses a 32-bit sequence number, as well as a I6-bit age field to identify LSAs. Recall that an OSPF node broadcasts LSAs for its attached links periodically, when a link cost to a neighbor changes, or when a link goes up/down. LSA sequence numbers are used to detect duplicate LSAs, but also serve a second important function in OSPF. With flooding, it is possible for an LSA generated by the source at time t to arrive after a newer LSA that was generated by the same source at time t + δ. The sequence numbers used by the source node allow an older LSA to be distinguished from a newer LSA. The age field serves a purpose similar to that of a TTL value. The initial age field value is set to zero and is incremented at each hop as it flooded, and is also incremented as it sits in a router's memory waiting to be flooded. Although we have only briefly described the LSA flooding algorithm here, we note that designing LSA broadcast protocols can be very tricky business indeed. [Perlman 1999] describe an incident in which incorrectly transmitted LSAs by two malfunctioning routers caused an early version of an LSA flooding algorithm to take down the entire ARPAnet.

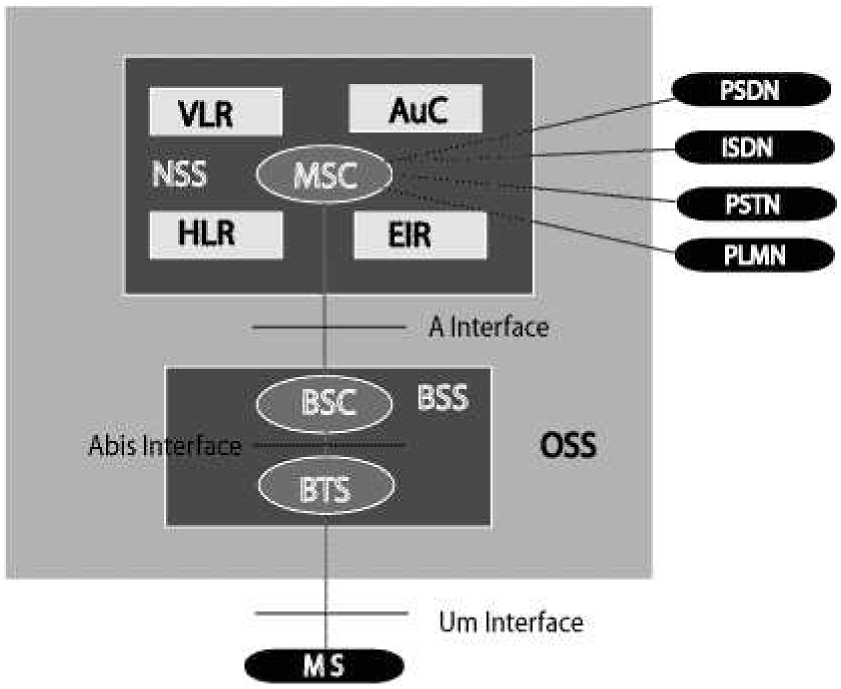
Module 4

7. a. Show the components of GSM 2G cellular network architecture with a diagram. (7)

A GSM network comprises of many functional units. These functions and interfaces are explained in this chapter. The GSM network can be broadly divided into:

- The Mobile Station (MS)
- The Base Station Subsystem (BSS)
- The Network Switching Subsystem (NSS)
- The Operation Support Subsystem (OSS)

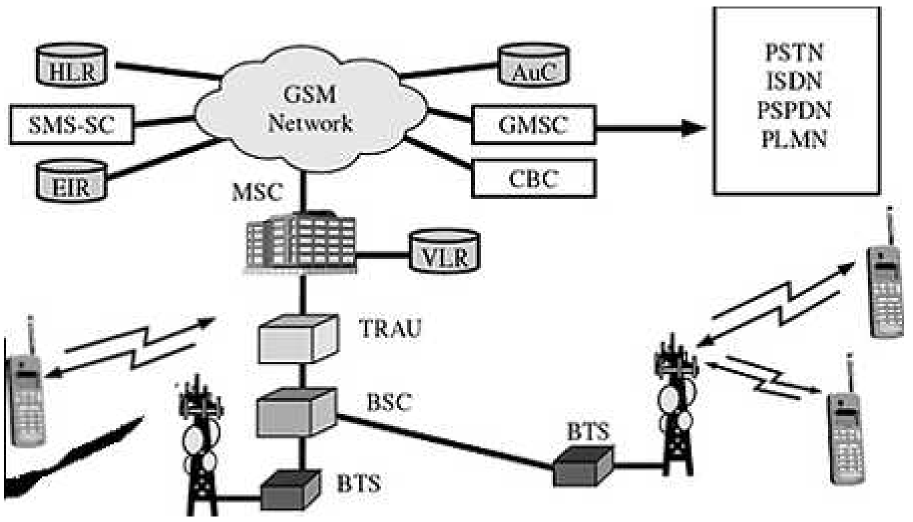
Given below is a simple pictorial view of the GSM architecture.



The additional components of the GSM architecture comprise of databases and messaging systems functions:

- Home Location Register (HLR)
- Visitor Location Register (VLR)
- Equipment Identity Register (EIR)
- Authentication Center (AuC)
- SMS Serving Center (SMS SC)
- Gateway MSC (GMSC)
- Chargeback Center (CBC)
- Transcoder and Adaptation Unit (TRAU)

The following diagram shows the GSM network along with the added elements:



The MS and the BSS communicate across the Um interface. It is also known as the air interface or the radio link. The BSS communicates with the Network Service Switching (NSS) center across the A interface.

GSM network areas

In a GSM network, the following areas are defined:

- Cell : Cell is the basic service area; one BTS covers one cell. Each cell is given a Cell Global Identity (CGI), a number that uniquely identifies the cell.
- Location Area : A group of cells form a Location Area (LA). This is the area that is paged when a subscriber gets an incoming call. Each LA is assigned a Location Area Identity (LAI). Each LA is served by one or more BSCs.
- MSC/VLR Service Area : The area covered by one MSC is called the MSC/VLR service area.
- PLMN : The area covered by one network operator is called the Public Land Mobile Network (PLMN). A PLMN can contain one or more MSCs.

---

b. Illustrate the steps involved in mobile IP registration with home agent. (5)

Mobile IP Registration

When the mobile node receives an agent advertisement, the mobile node registers through the foreign agent, even when the mobile node might be able to acquire its own co-located care-of address. This feature enables sites to restrict access to mobility services. Through agent advertisements, mobile nodes detect when they have moved from one subnet to another.

Mobile IP registration provides a flexible mechanism for mobile nodes to communicate their current reachability information to their home agent. The registration process enables mobile nodes to perform the following tasks:

- Request forwarding services when visiting a foreign network
- Inform their home agent of their current care-of address
- Renew a registration that is due to expire
- Deregister when they return home

Registration messages exchange information between a mobile node, a foreign agent, and the home agent. Registration creates or modifies a mobility binding at the home agent, associating the mobile node's home address with its care-of address for the specified lifetime.

The registration process also enables mobile nodes to:

- Register with multiple foreign agents
- Deregister specific care-of addresses while retaining other mobility bindings
- Discover the address of a home agent if the mobile node is not configured with this information

Mobile IP defines the following registration processes for a mobile node:

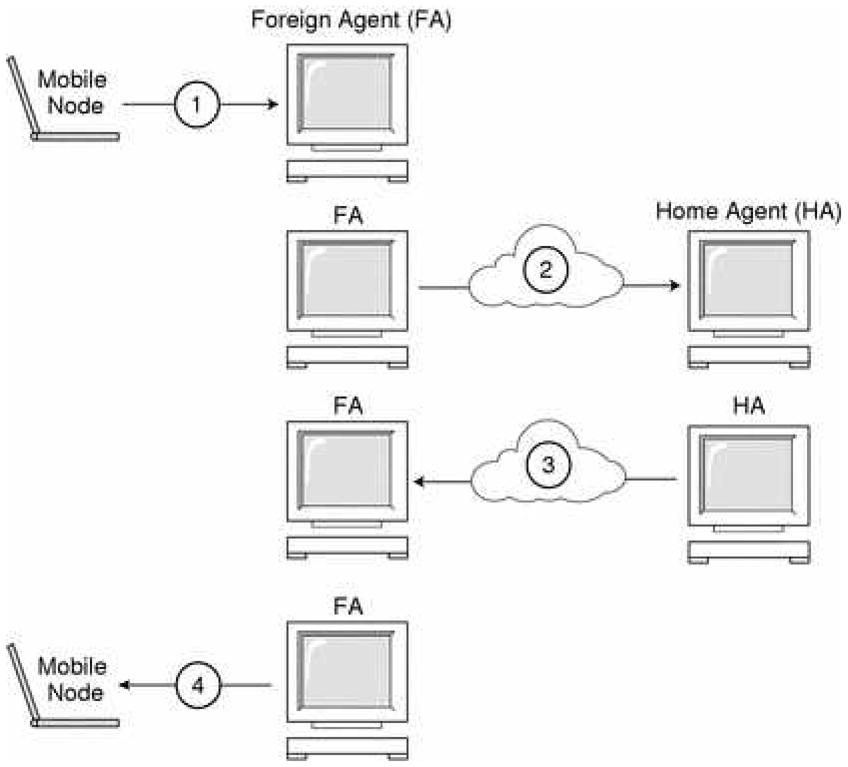
- If a mobile node is registering a foreign agent care-of address, the mobile node registers using that foreign agent.
- If a mobile node is using a co-located care-of address, and receives an agent advertisement from a foreign agent on the link on which it is using this care-of address, the mobile node registers using that foreign agent (or another foreign agent on this link).
- If a mobile node uses a co-located care-of address, the mobile node registers directly with its home agent.
- If a mobile node returns to its home network, the mobile node deregisters with its home agent.

These registration processes involve the exchange of registration requests and registration reply messages. When registering using a foreign agent, the registration process takes the following steps, which the subsequent illustration depicts:

1. The mobile node sends a registration request to the prospective foreign agent to begin the registration process.

2. The foreign agent processes the registration request and then relays it to the home agent.
3. The home agent sends a registration reply to the foreign agent to grant or deny the request.
4. The foreign agent processes the registration reply and then relays it to the mobile node to inform it of the disposition of its request.

Figure 1-4 Mobile IP Registration Process



When the mobile node registers directly with its home agent, the registration process requires only the following steps:

- The mobile node sends a deregistration request to the home agent.
- The home agent sends a registration reply to the mobile node, granting or denying the request.

c. Write a note on mobile IP. (4)

Introduction

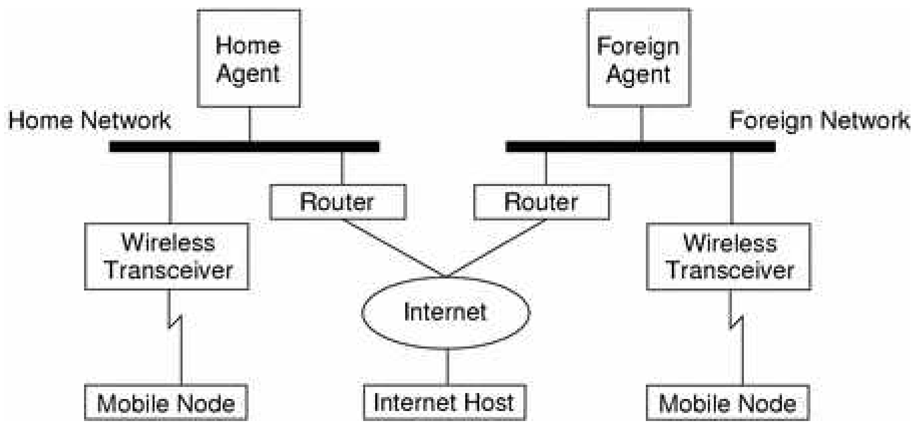
Current versions of the Internet Protocol (IP) assume that the point at which a computer attaches to the Internet or a network is fixed and its IP address identifies the network to which it is attached. Datagrams are sent to a computer based on the location information contained in the IP address.

If a mobile computer, or mobile node, moves to a new network while keeping its IP address unchanged, its address does not reflect the new point of attachment. Consequently, existing routing protocols cannot route datagrams to the mobile node correctly. In this situation, you must reconfigure the mobile node with a different IP address representative of its new location, which is a cumbersome process. Thus, under the current Internet Protocol, if the mobile node moves without changing its address, it loses routing; but if it does change its address, it loses connections.

Mobile IP solves this problem by allowing the mobile node to use two IP addresses: a fixed home addressand a care-of addressthat changes at each new point of attachment. Mobile IP enables a computer to roam freely on the Internet or an organization's network while still maintaining the same home address. Consequently, computing activities are not disrupted when the user changes the computer's point of attachment to the Internet or an organization's network. Instead, the network is updated with the new location of the mobile node. See Glossary for definitions of terms associated with Mobile IP.

The following figure illustrates the general Mobile IP topology.

Figure 1–1 Mobile IP Topology



Using the previous illustration's Mobile IP topology, the following scenario shows how a datagram moves from one point to another within the Mobile IP framework.

1. The Internet host sends a datagram to the mobile node using the mobile node's home address (normal IP routing process).
2. If the mobile node is on its home network, the datagram is delivered through the normal IP process to the mobile node. Otherwise, the home agent picks up the datagram.
3. If the mobile node is on a foreign network, the home agent forwards the datagram to the foreign agent.
4. The foreign agent delivers the datagram to the mobile node.
5. Datagrams from the mobile node to the Internet host are sent using normal IP routing procedures. If the mobile node is on a foreign network, the packets are delivered to the foreign agent. The foreign agent forwards the datagram to the Internet host.

In the case of wireless communications, the illustrations depict the use of wireless transceivers to transmit the datagrams to the mobile node. Also, all datagrams between the Internet host and the mobile node use the mobile node's home address regardless of whether the mobile node is on a home or foreign network. The care-of address is used only for communication with mobility agents and is never seen by the Internet host.

Mobile IP Functional Entities

Mobile IP introduces the following new functional entities:

- Mobile Node (MN)–Host or router that changes its point of attachment from one network to another.
- Home Agent (HA)–Router on a mobile node's home network that intercepts datagrams destined for the mobile node, and delivers them through the care-of address. The home agent also maintains current location information for the mobile node.
- Foreign Agent (FA)–Router on a mobile node's visited network that provides routing services to the mobile node while the mobile node is registered.

How Mobile IP Works

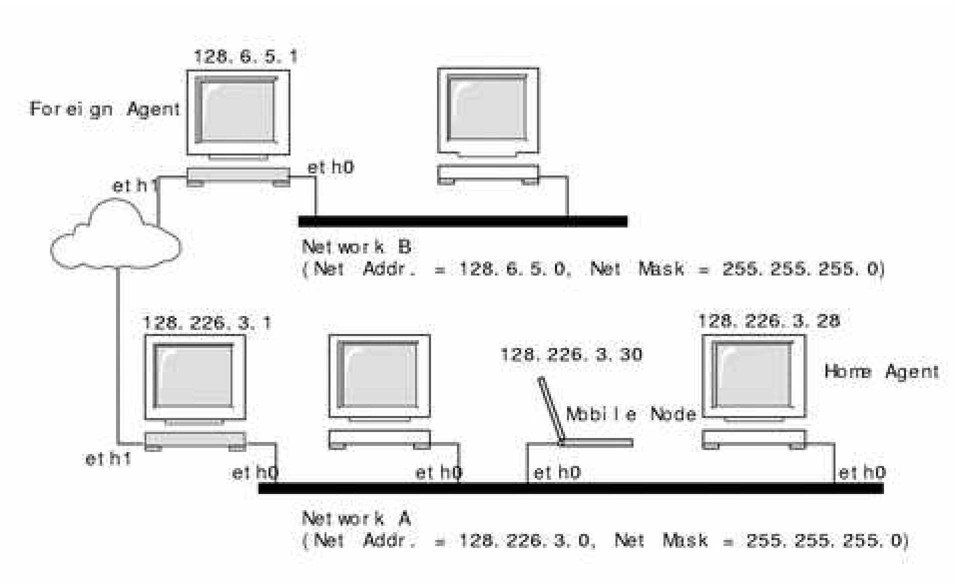
Mobile IP enables routing of IP datagrams to mobile nodes. The mobile node's home address always identifies the mobile node, regardless of its current point of attachment to the Internet or an organization's network. When away from home, a care-of address associates the mobile node with its home address by providing information about the mobile node's current point of attachment to the Internet or an organization's network.

Mobile IP uses a registration mechanism to register the care-of address with a home agent.

The home agent redirects datagrams from the home network to the care-of address by constructing a new IP header that contains the mobile node's care-of address as the destination IP address. This new header then encapsulates the original IP datagram, causing the mobile node's home address to have no effect on the encapsulated datagram's routing until it arrives at the care-of address. This type of encapsulation is also called tunneling. After arriving at the care-of address, each datagram is de-encapsulated and then delivered to the mobile node.

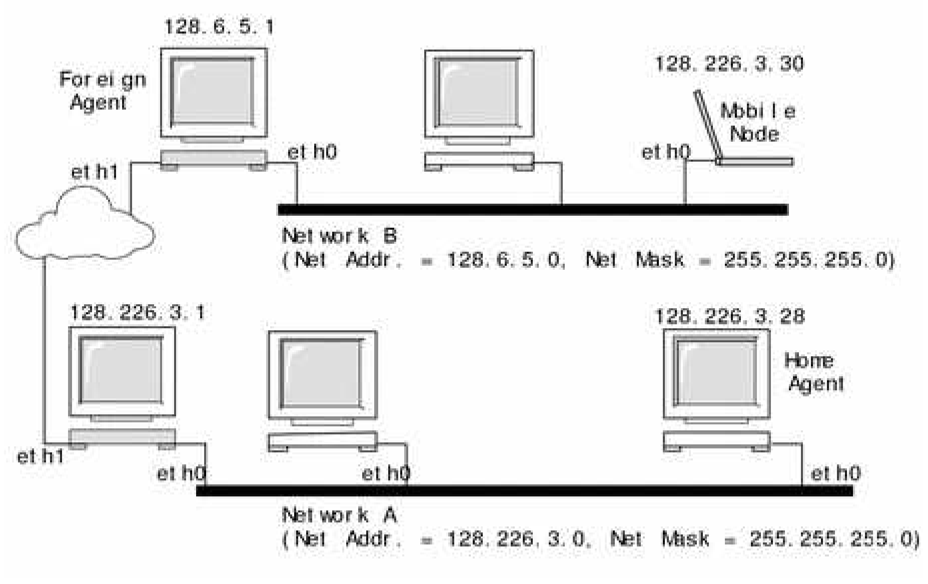
The following illustration shows a mobile node residing on its home network, Network A, before the mobile node moves to a foreign network, Network B. Both networks support Mobile IP. The mobile node is always associated with its home network by its permanent IP address, 128.226.3.30. Though Network A has a home agent, datagrams destined for the mobile node are delivered through the normal IP process.

Figure 1–2 Mobile Node Residing on Home Network



The following illustration shows the mobile node moving to a foreign network, Network B. Datagrams destined for the mobile node are intercepted by the home agent on the home network, Network A, encapsulated, and sent to the foreign agent on Network B. Upon receiving the encapsulated datagram, the foreign agent strips off the outer header and delivers the datagram to the mobile node visiting Network B.

Figure 1–3 Mobile Node Moving to a Foreign Network



The care-of address might belong to a foreign agent, or might be acquired by the mobile node through Dynamic Host Configuration Protocol (DHCP) or Point-to-Point Protocol (PPP). In the latter case, a mobile node is said to have a co-located care-of address.

The mobile node uses a special registration process to keep its home agent informed about its current location. Whenever a mobile node moves from its home network to a foreign network, or from one foreign network to another, it chooses a foreign agent on the new network and uses it to forward a registration message to its home agent.

Mobility agents (home agents and foreign agents) advertise their presence using agent advertisement messages. A mobile node can optionally solicit an agent advertisement message from any locally attached mobility agents through an agent solicitation message. A mobile node receives these agent advertisements and determines whether they are on its home network or a foreign network.

When the mobile node detects that it is located on its home network, it operates without mobility services. If returning to its home network from being registered elsewhere, the mobile node deregisterswith its home agent.

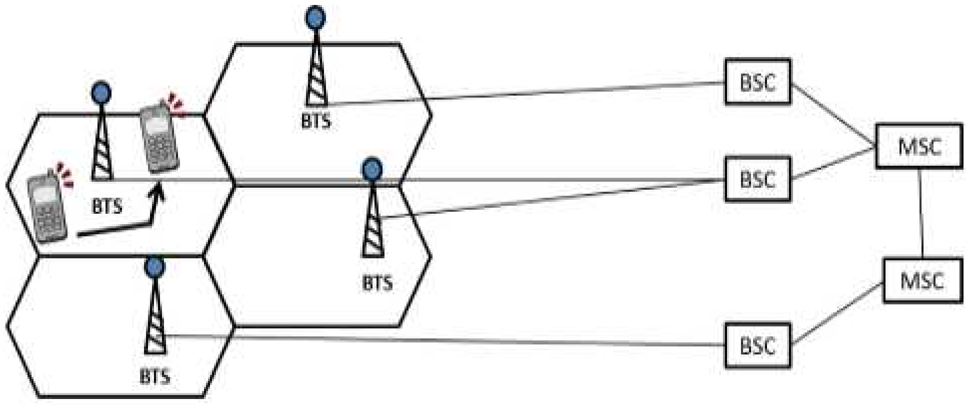
8. a. Define Handoff, Explain the steps accomplishing a handoff. (7)

- Handoff (or handover) is a control process initiated when a mobile moves from its current cell to its neighboring cell.

- A user of a mobile phone will be moving continuously. In such a situation, the mobile connection should also remain intact especially if the user is currently using the phone.
- This transfer of connection from one cell to another should be quick and in such a manner that user doesn't actually realize that a handoff has happened.
- There are four basic types of handoffs in GSM network:

a) Intra-cell handover:

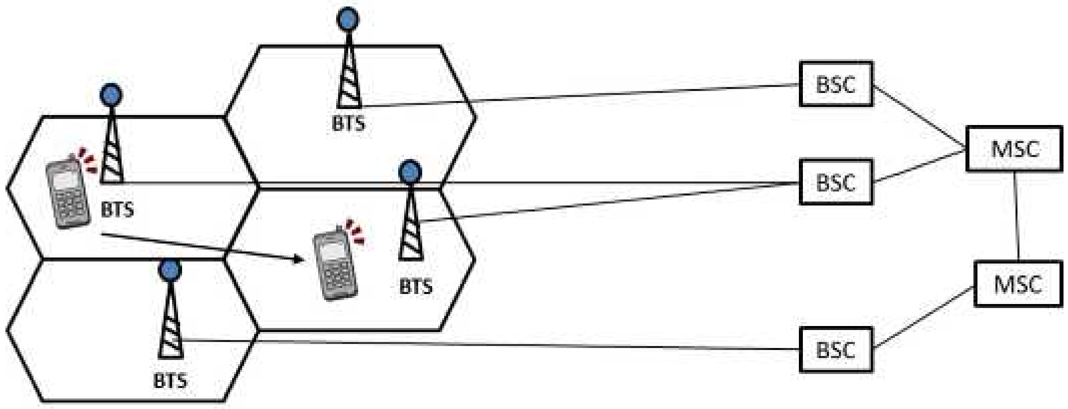
  - Such a kind of handover is performed to optimize the traffic load in the cell or to improve quality of a connection by changing carrier frequency.



Intra-cell handover

b) Inter-cell handover:

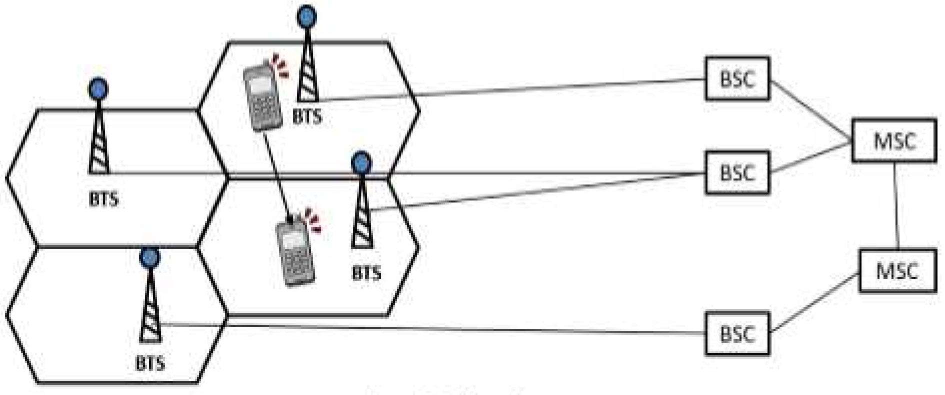
- It is also known as Intra-BSC handover.
- Here the mobile moves from one cell to another but remains within the same BSC (Base station controller).
- Here the BSC handles the handover process



Intra-BSC handover

c) Inter-BSC handover:

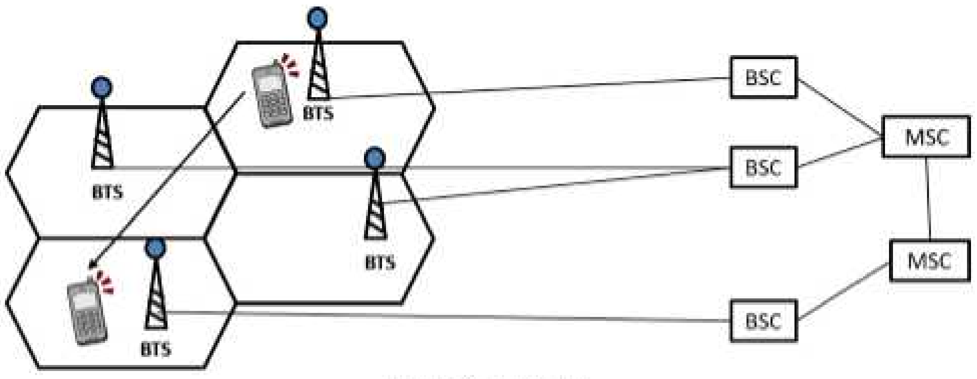
- It is also called as Intra-MSC handover.
- As BSC can control only a limited number of cells, we might usually need to transfer a mobile from one BSC to another BSC.
- Here the MSC handles the handover process.



Inter-BSC handover

d) Inter-MSC handover:

- It occurs when a mobile moves from one MSC region to another MSC.
- MSC cover a large area. It can be imagined as a handover from Maharashtra MSC to Gujarat MSC while travelling.



Inter-MSC handover

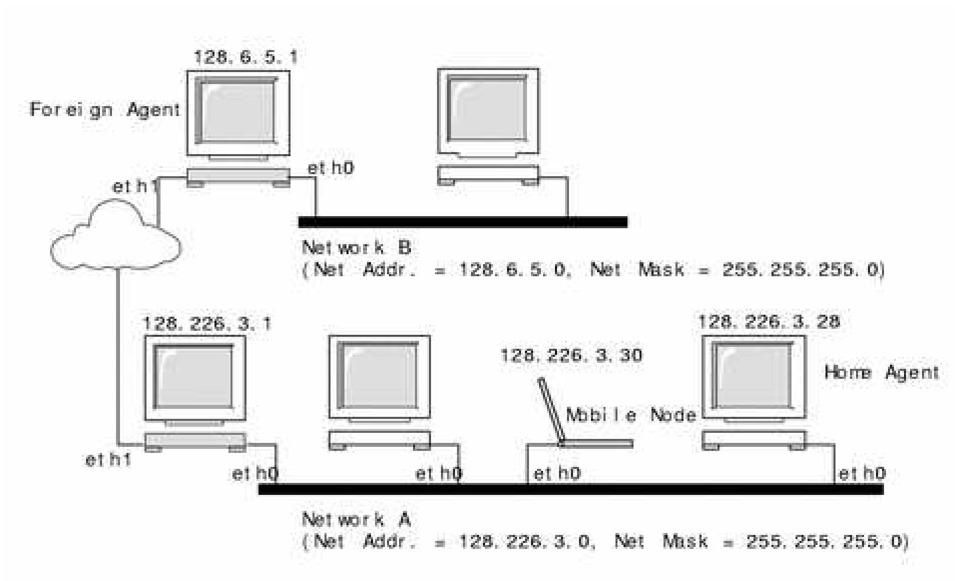
b. Bring out the mechanism of direct routing to mobile node in mobility management. (6)

Mobile IP enables routing of IP datagrams to mobile nodes. The mobile node's home address always identifies the mobile node, regardless of its current point of attachment to the Internet or an organization's network. When away from home, a care-of address associates the mobile node with its home address by providing information about the mobile node's current point of attachment to the Internet or an organization's network. Mobile IP uses a registration mechanism to register the care-of address with a home agent.

The home agent redirects datagrams from the home network to the care-of address by constructing a new IP header that contains the mobile node's care-of address as the destination IP address. This new header then encapsulates the original IP datagram, causing the mobile node's home address to have no effect on the encapsulated datagram's routing until it arrives at the care-of address. This type of encapsulation is also called tunneling. After arriving at the care-of address, each datagram is de-encapsulated and then delivered to the mobile node.

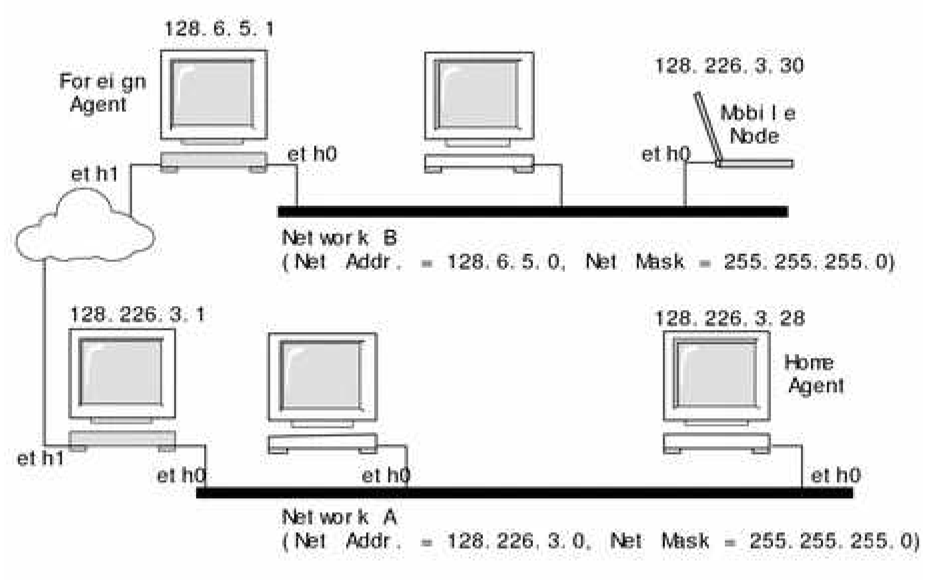
The following illustration shows a mobile node residing on its home network, Network A, before the mobile node moves to a foreign network, Network B. Both networks support Mobile IP. The mobile node is always associated with its home network by its permanent IP address, 128.226.3.30. Though Network A has a home agent, datagrams destined for the mobile node are delivered through the normal IP process.

Figure 1–2 Mobile Node Residing on Home Network



The following illustration shows the mobile node moving to a foreign network, Network B. Datagrams destined for the mobile node are intercepted by the home agent on the home network, Network A, encapsulated, and sent to the foreign agent on Network B. Upon receiving the encapsulated datagram, the foreign agent strips off the outer header and delivers the datagram to the mobile node visiting Network B.

Figure 1–3 Mobile Node Moving to a Foreign Network



The care-of address might belong to a foreign agent, or might be acquired by the mobile node through Dynamic Host Configuration Protocol (DHCP) or Point-to-Point Protocol (PPP). In the latter case, a mobile node is said to have a co-located care-of address.

The mobile node uses a special registration process to keep its home agent informed about its current location. Whenever a mobile node moves from its home network to a foreign network, or from one foreign network to another, it chooses a foreign agent on the new network and uses it to forward a registration message to its home agent.

Mobility agents (home agents and foreign agents) advertise their presence using agent advertisement messages. A mobile node can optionally solicit an agent advertisement message from any locally attached mobility agents through an agent solicitation message. A mobile node receives these agent advertisements and determines whether they are on its home network or a foreign network.

When the mobile node detects that it is located on its home network, it operates without mobility services. If returning to its home network from being registered elsewhere, the mobile node deregisters with its home agent.

c. Compare the 4G LTE standard to 3G systems. (3)

Comparison chart

3G versus 4G comparison chart		
	3G	4G
Data Throughput	Up to 3.1Mbps with an average speed range between 0.5 to 1.5 Mbps	Practically speaking, 2 to 12 Mbps (Telstra in Australia claims up to 40 Mbps) but potential estimated at a range of 100 to 300 Mbps.
Peak Upload Rate	5 Mbps	500 Mbps
Switching Technique	packet switching	packet switching, message switching
Network Architecture	Wide Area Cell Based	Integration of wireless LAN and Wide area.
Services And Applications	CDMA 2000, UMTS, EDGE etc	Wimax2 and LTE-Advance
Forward error correction (FEC)	3G uses Turbo codes for error correction.	Concatenated codes are used for error corrections in 4G.
Peak Download Rate	100 Mbps	1 Gbps
Frequency Band	1.8 – 2.5 GHz	2 – 8 GHz

Module-5

9. a. Elaborate the features of streaming stored video. (3)

   Streaming media is multimedia that is constantly received by and presented to an end-user while being delivered by a provider. The verb "to stream" refers to the process of delivering or obtaining media in this manner; the term refers to the delivery method of the medium, rather than the medium itself, and is an alternative to file downloading, a process in which the end-user obtains the entire file for the content before watching or listening to it.

A client end-user can use their media player to start playing the data file (such as a digital file of a movie or song) before the entire file has been transmitted. Distinguishing delivery method from the media distributed applies specifically to telecommunications networks, as most of the delivery systems are either inherently streaming (e.g. radio, television, streaming apps) or inherently non-streaming (e.g. books, video cassettes, audioCDs). For example, in the 1930s, elevator music was among the earliest popularly available streaming media; nowadays Internet television is a common form of streamed media. The term "streaming media" can apply to media other than video and audio such as live closed captioning, ticker tape, and real-time text, which are all considered "streaming text".

The term "streaming" was first used for tape drives made by Data Electronics Inc. for drives meant to slowly ramp up and run for the entire track; the slow ramp times resulted in lower drive costs, making a more competitive product. "Streaming" was applied in the early 1990s as a better description for video on demand on IP networks; at the time such video was usually referred to as "store and forward video" which was misleading nomenclature.

Live streaming is the delivery of Internet content in real-time, as events happen, much as live television broadcasts its contents over the airwaves via a television signal. Live internet streaming requires a form of source media (e.g. a video camera, an audio interface, screen capture software), an encoder to digitize the content, a media publisher, and a content delivery network to distribute and deliver the content. Live streaming does not need to be recorded at the origination point, although it frequently is.

As of 2017, "streaming" generally refers to the situation where a user watches digital video content or listens to digital audio content on a computer screen and speakers (ranging from a smartphone, through a desktop computer to a large-screen home entertainment system) over the Internet. With streaming content, the user does not have to download the entire digital video or digital audio file before they start to play it.

There are challenges with streaming content on the Internet. If the user does not have enough bandwidth in their Internet connection, they may experience stops in the content and some users may not be able to stream certain content due to not having compatible computer or software systems.

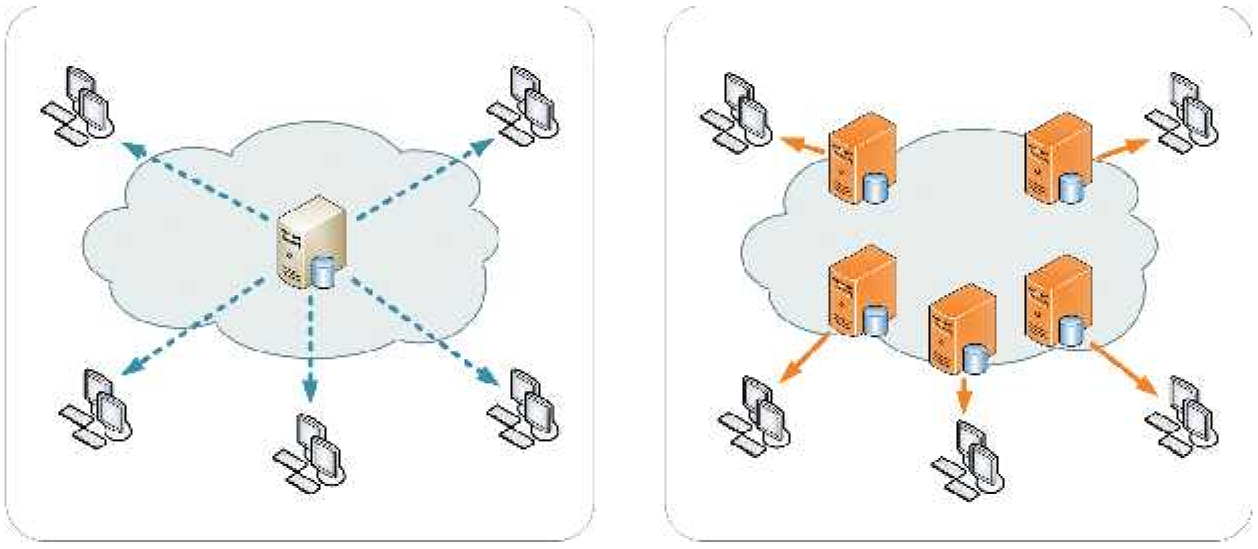
Some popular streaming services are the video sharing websiteYouTube; Twitch and Mixer, which live stream the playing of video games; Netflix, which streams movies and TV shows; and Spotify and Apple Music, which stream music.

b. With a neat diagram, explain the CDN operation. (8)

A content delivery network or content distribution network (CDN) is a geographically distributed network of proxy servers and their data centers. The goal is to distribute service spatially relative to end-users to provide high availability and high performance. CDNs serve a large portion of the Internet content today, including web objects (text, graphics and scripts), downloadable objects (media files, software, documents), applications (e-commerce, portals), live streaming media, on-demand streaming media, and social networks.

CDNs are a layer in the internet ecosystem. Content owners such as media companies and e-commerce vendors pay CDN operators to deliver their content to their end users. In turn, a CDN pays ISPs, carriers, and network operators for hosting its servers in their data centers.

CDN is an umbrella term spanning different types of content delivery services: video streaming, software downloads, web and mobile content acceleration, licensed/managed CDN, transparent caching, and services to measure CDN performance, load balancing, multi-CDN switching and analytics and cloud intelligence. CDN vendors may cross over into other industries like security and WAN optimization.



c. Summarize the limitations of Best-effort IP service. (5)

Best-effort service can lead to packet loss, excessive end-to-end delay, and packet jitter.

Packet Loss
Consider one of the UDP segments generated by our Internet phone applicatio. The UDP segment is encapsulated in an IP datagram. As the datagram wanders through the network, it passes through buffers in the routers in order to access outbound links. It is possible that one or more of the buffers in the route from sender to receiver is full and cannot admit the IP datagram. The IP datagram is discarded, never to arrive at the receiving application.

Loss could be eliminated by sending packets the packets oer TCP rather than over UDP. Recall that TCP retransmits packets that do not arrive at the destination. Retransmission mechanisms are often considered unacceptable for interacitve real-time audio applications such as Internet phone, because they increase end-to-end delay.

End-to-End Delay
End-to-end delay is the accumulation of transmission, porcessing, and queuing delays in routers; propagation delays in the links; and end-system processing delays.

Packet Jitter
A crucial component of end-to-end delay is the random queuing delays in the routers. Boecause of these varying delays within the network, the time from when a packet is generated at the source until it is received at the receiver can fluctuate from packet to packet. This phenomenon is called jitter.

Jitter can often be removed by using sequence numbers, timestamps, and a playout delay.

10. a. Explain the diffserv internet architecture. (5)

Differentiated Services Architecture (Diff-Serv)

Diff-Serv is the product of an IETF working group that has defined a more scalable way to apply IP QoS. It has particular relevance to service provider networks. Diff-Serv minimizes signaling and concentrates on aggregated flows and per hop behaviour (PHB) applied to a network-wide set of traffic classes. Arriving flows are classified according to pre-determined rules, which aggregate many application flows into a limited and manageable set (perhaps 2 to 8) of class flows.

Traffic entering the network domain at the edge router is first classified for consistent treatment at each transit router inside the network. Treatment will usually be applied by separating traffic into different queues according to the class of traffic, so that high-priority packets can be assigned the appropriate priority level at an output port.

DiffServ approach separates the classification and queuing functions. Packets carry self-evident priority marking in the Type-of-Service byte inside packet headers. (ToS byte is part of the legacy IP architecture.) IP Precedence (IPP) defines eight priority levels. DiffServ, its emerging replacement, reclaims the entire ToS byte to define up to a total of 256 levels. The priority value is interpreted as an index into a Per-Hop Behavior (PHB) that defines the way a single network node should treat a packet marked with this value, so that it will provide consistent multi-hop service. In many cases, PHBs are implemented using some of the queuing disciplines mentioned earlier. This method allows for an efficient index classification that is considered to be highly scalable and best suited for backbone use. However, translating PHBs into end-to-end QoS is not a trivial task. Moreover, inter-domain environments may require a concept known as "bandwidth brokerage," which is still in the early research stage.

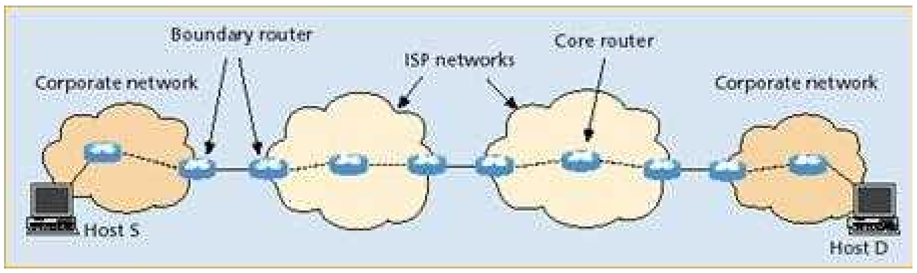


Figure 6 - End-to-end transport from host S to host D under the DiffServ architecture

DiffServ outlines an initial architectural philosophy that serves as a framework for inter-provider agreements and makes it possible to extend QoS beyond a single network domain. The DiffServ framework is more scalable than IntServ because it handles flow aggregates and minimizes signaling, thus avoiding the complexity of per-flow soft states at each node. Diff-Serv will likely be applied most commonly in enterprise backbones and in service provider networks.

There will be domains where IntServ and DiffServ coexist, so there is a need to interwork them at boundaries. This interworking will require a set of rules governing the aggregation of individual flows into class flows suitable for transport through a Diff-Serv domain. Several draft interworking schemes have been submitted to the IETF.

An Overview of DiffServ

To provide QoS support, a network must somehow allow for controlled unfairness in the use of its resources. Controlling to a granularity as fine as a flow of data requires advanced signaling protocols. By recognizing that most of the data flows generated by different applications can be ultimately classified into a few general categories (i.e., traffic classes), the DiffServ architecture aims at providing simple and scalable service differentiation. It does this by discriminating and treating the data flows according to their traffic class, thus providing a logical separation of the traffic in the different classes.

In DiffServ, scalability and flexibility are achieved by following a hierarchical model for network resource management:

Interdomain resource management:

Unidirectional service levels, and hence traffic contracts, are agreed at each boundary point between a customer and a provider for the traffic entering the provider network.

Intradomain resource management:

> The service provider is solely responsible for the configuration and provisioning of resources within its domain (i.e., the network). Furthermore, service policies are also left to the provider.

At their boundaries, service providers build their offered services with a combination of traffic classes (to provide controlled unfairness), traffic conditioning (a function that modifies traffic characteristics to make it conform to a traffic profile and thus ensure that traffic contracts are respected), and billing (to control and balance service demand). Provisioning and partitioning of both boundary and interior resources are the responsibility of the service provider and, as such, outside the scope of DiffServ. For example, DiffServ does not impose either the number of traffic classes or their characteristics on a service provider.

Although traffic classes are nominally supported by interior routers, DiffServ does not impose any requirement on interior resources and functionalities. For example, traffic conditioning (i.e., metering, marking, shaping, or dropping) in the interior of a network is left to the discretion of the service providers.

If each packet conveyed across a service provider's network simply carries in its header an identification of the traffic class (called a DS codepoint) to which it belongs, the network can easily provide a different level of service to each class. It does this by appropriately treating the corresponding packets, say, by selecting the appropriate per-hop behavior (PHB) for each packet. In both IPv4 and IPv6, the traffic class is denoted by use of the DS header field.

It must be noted that DiffServ is based on local service agreements at customer/provider boundaries. Therefore, end-to-end services will be built by concatenating such local agreements at each domain boundary along the route to the final destination. The concatenation of local services to provide meaningful end-to-end services is still an open research issue.

The net result of the DiffServ approach is that per-flow state is avoided within the network, since individual flows are aggregated in classes.

Benefits of DiffServ

The DiffServ architecture is an elegant way to provide much needed service discrimination within a commercial network. Customers willing to pay more will see their applications receive better service than those paying less. This scheme exhibits an "auto-funding" property: "popular" traffic classes generate more revenues, which can be used to increase their provisioning.

A traffic class is a predefined aggregate of traffic. Compared with the aggregate of flows described earlier, traffic classes in DiffServ are accessible without signaling, which means they are readily available to applications without any setup delay. Consequently, traffic classes can provide qualitative or relative services to applications that cannot express their requirements quantitatively. This conforms to the original design philosophy of the Internet. An example of qualitative service is "traffic offered at service level A will be delivered with low latency," while a relative service could be "traffic offered at service level A will be delivered with higher probability than traffic offered at service level B." Quantitative services can also be provided by DiffServ. A quantitative service might be "90 percent of in-profile traffic offered at service level C will be delivered."

Since the provisioning of traffic classes is left to the provider's discretion, this provisioning can, and in the near future will, be performed statically and manually. Hence, existing management tools and protocols can be used to that end. However, this does not rule out the possibility of more automatic procedures for provisioning.

The only functionality actually imposed by DiffServ in interior routers is packet classification. This classification is simplified from that in RSVP because it is based on a single IP header field containing the DS codepoint, rather than multiple fields from different headers. This has the potential of allowing functions performed on every packet, such as traffic policing or shaping, to be done at the boundaries of domains, so forwarding is the main operation performed within the provider network.

Another advantage of DiffServ is that the classification of the traffic, and the subsequent selection of a DS codepoint for the packets, need not be performed in the end systems. Indeed, any router in the stub network where the host resides, or the ingress router at the boundary between the stub and provider networks, can be configured to classify (on a per-flow basis), mark, and shape the traffic from the hosts. Such routers are the only points where per-flow classification may occur, which does not pose any problem because they are at the edge of the Internet, where flow concentration is low. The potential noninvolvement of end systems, and the use of existing and widespread management tools and protocols allows swift and incremental deployment of the DiffServ architecture.

b. Describe the leaky bucket policing mechanism. (6)

policing, the regulation of the rate at which a flow is allowed to inject packets into the network, as one of the cornerstones of any QoS architecture. But what aspects of a flow's packet rate should be policed? We can identify three important policing criteria, each differing from the other according to the time scale over which the packet flow is policed:

- Average rate. The network may wish to limit the long-term average rate (packets per time interval) at which a flow's packets can be sent into the network. A crucial issue here is the interval of time over which the average rate will be policed. A flow whose average rate is limited to 100 packets per second is more constrained than a source that is limited to 6000 packets per minute, even though both have the same average rate over a long enough interval of time. For example, the latter constraint would allow a flow to send 1000 packets in a given second-long interval of time (subject to the constraint that the rate be less that 6000 packets over a minute-long interval containing these 1000 packets), while the former constraint would disallow this sending behavior.

- Peak rate. While the average rate constraint limits the amount of traffic that can be sent into the network over a relatively long period of time, a peak rate constraint limits the maximum number of packets that can be sent over a shorter period of time. Using our example above, the network may police a flow at an average rate of 6000 packets per minute, while limiting the flow's peak rate to 1500 packets per second.

- Burst size. The network may also wish to limit the maximum number of packets (the "burst" of packets) that can be sent into the network over a extremely short interval of time. In the limit as the interval length approaches zero, the burst size limits the number of packets that can be instantaneously sent into the network. While it is physically impossible to instantaneously send multiple packets into the network (after all, every link has a physical transmission rate that can not be exceeded!), the abstraction of a maximum burst size is a useful one.

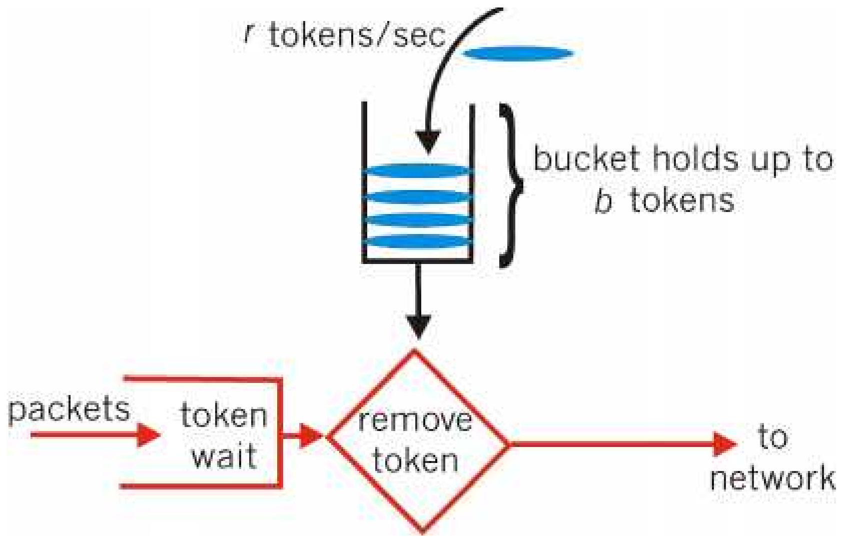


Figure 6.6-7: The Leaky Bucket Policer

The leaky bucket (also call a token bucket) mechanism is an abstraction that can be used to characterize these policing limits. As shown in Figure 6.6-7, a leaky bucket consists of a bucket that can hold up to b tokens. Tokens are added to this bucket as follows. New tokens, which may potentially be added to the bucket, are always being generated at a rate of r tokens per second. (We assume here for simplicity that the unit of time is a second.) If the bucket is filled with less thatb tokens when a token is generated, the newly generated token is added to the bucket; otherwise the newly generated token is ignored, and the token bucket remains full with b tokens.

Let us now consider how is the leaky bucket can be used to police a packet flow. Suppose before a packet is transmitted into the network, it must first remove a token from the token bucket. If the token bucket is empty, the packet must wait for a token. (An alternative is for the packet to be dropped, although we will not consider that option here.) Let us now consider how this behavior polices a traffic flow. Because there can be at most b tokens in the bucket, the maximum burst size for a leaky-bucket-policed flow is b packets. Furthermore, because the token generation rate is r, the maximum number of packets that can enter the network of any interval of time of length t is rt+b. Thus, the token generation rate, r, serves to limit the long term average rate at which packet can enter the network. It is also possible to use leaky buckets (specifically, two leaky buckets in series) to police a flow's peak rate in addition to the long-term average rate; see the homework problems at the end of this Chapter.

Leaky Bucket + Weighted Fair Queuing => Provable Maximum Delay in a Queue

We will see that both leaky bucket policing and WFQ scheduling will play an important role. Let us thus close this section by considering a router's output that multiplexes n flows, each policed by a leaky bucket with parameters b_i andr_i, i = 1,...,n, using WFQ scheduling. We assume that each flow is treated as a separate class by the WFQ scheduler, as shown in Figure 6.6-8.

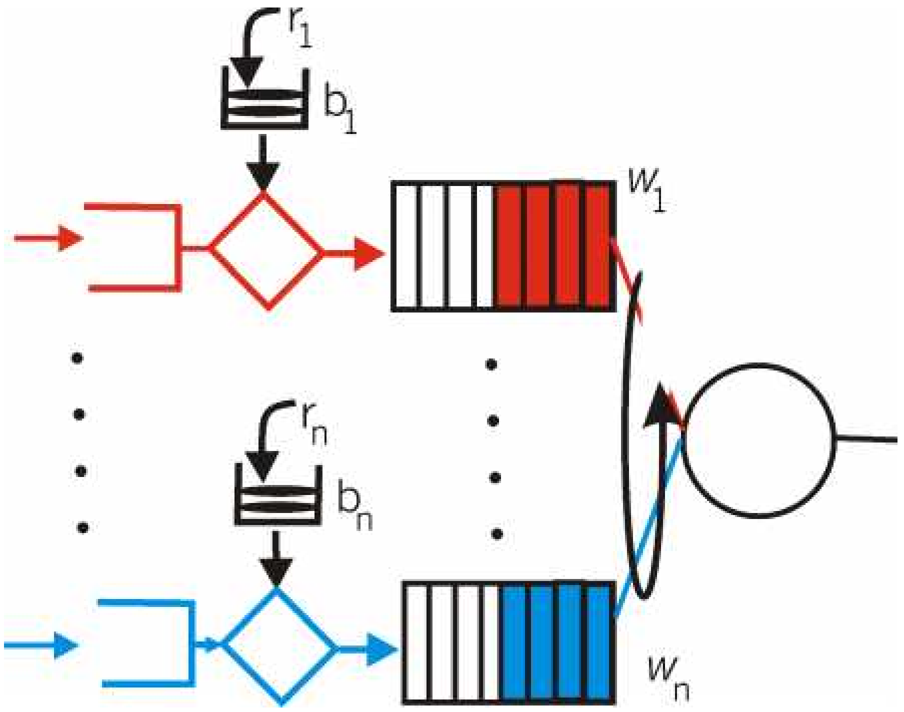


Figure 6.6-8:n multiplexed leaky bucket flows with WFQ scheduling

Recall from our discussion of WFQ that each flow is guaranteed to receive a share of the link bandwidth equal to at least $R \cdot w_i/(\ \ w_j)$, where R is the transmission rate of the link in packets/sec. What then is the maximum delay that a packet will experience while waiting for service in the WFQ (i.e., after passing through the leaky bucket)? Let us focus on flow 1. Suppose that flow 1's token bucket is initially full. A burst of b_1 packets then arrives to the leaky bucket policer for flow 1. These packets remove all of the tokens (without wait) from the leaky bucket and then join the WFQ waiting area for flow 1. Since these $b_1$packets are served at a rate of at least $R \cdot w_i/(\ \ w_j)$ packet/sec., the last of these packets will then have a maximum delay, d_{max}, until its transmission is completed, where

$d_{max}= \ b_1/\ (C \cdot w_i/(\ \ w_j))$

The justification of this formula is that if there are b_1 packets in the queue and packets are being serviced (removed) from the queue at a rate of at least $C \cdot w_i/(\ \ w_j)$ packets per second, then the amount of time until the last bit of the last packet is transmitted can not be more than $b_1/\ (C{\cdot}w_i/(\ \ w_j))$. A homework problem asks you prove that as long as $r_1<C{\cdot}w_i/(\ \ w_j)$, then$d_{max}$ is indeed the maximum delay that any packet in flow 1 will ever experience in the WFQ queue.

c. Discuss the round-robin and waited fair queuing scheduling mechanism. (5)

Under the round robin queuing discipline, packets are again sorted into classes, as with priority queuing. However, rather than there being a strict priority of service among classes, a round robin scheduler alternates service among the classes. In the simplest form of round robin scheduling, a class 1 packet is transmitted, followed by a class 2 packet, followed by a class 1 packet, followed by a class 2 packet, etc. A so-called work-conserving queuing discipline will never allow the link to remain idle whenever there are packets (of any class) queued for transmission. A work-conserving round robin discipline that looks for a packet of a given class but finds none will immediately check the next class in the round robin sequence.

Figure 6.6-5 illustrates the operating of a two-class round robin queue. In this example, packets 1, 2 and 4 belong to class one, and packets 3 and 5 belong to the second class. Packet 1 begins transmission immediately upon arrival at the output queue. Packets 2 and 3 arrive during the transmission of packet 1 and thus queue for transmission. After the transmission of packet 1, the link scheduler looks for a class-two packet and thus transmits packet 3. After the transmission of packet 3, the scheduler looks for a class-one packet and thus transmits packet 2. After the transmission of packet 2, packet 4 is the only queued packet; it is thus transmitted immediately after packet 2.

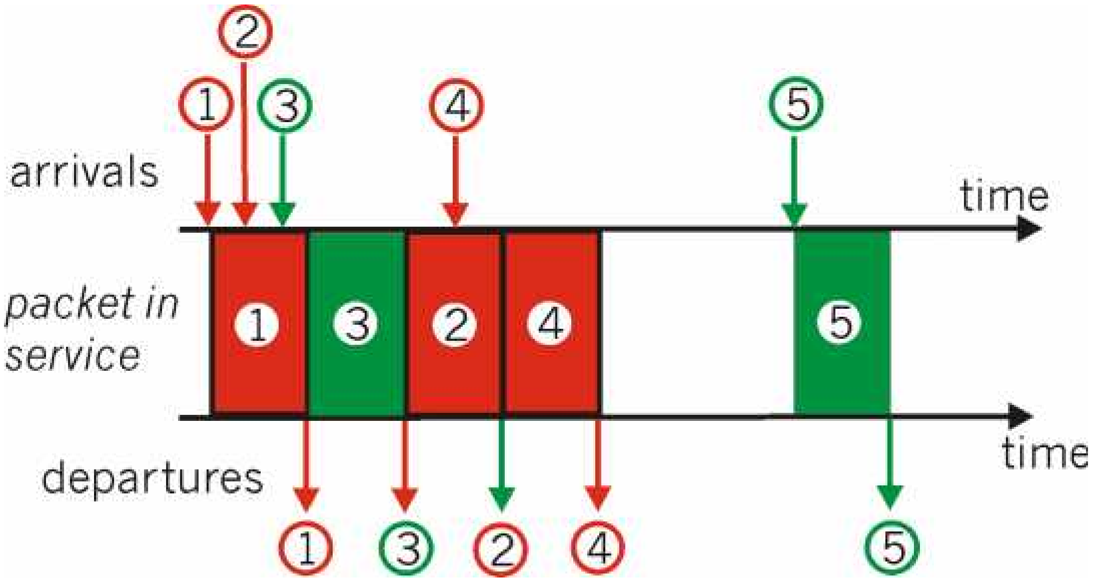


Figure 6.6-5: Operation of the two-class round robin queue

A generalized abstraction of round robin queuing that has found considerable use in QoS architectures is the so-called Weighted Fair Queuing (WFQ) discipline [Demers 90, Parekh 93]. WFQ is illustrated in Figure 6.6-6. Arriving packets are again classified and queued in the appropriate per-class waiting area. As in round robin scheduling, a WFQ scheduler will again serve classes in a circular manner - first serving class 1, then serving class 2, then serving class 3, and then (assuming there are three classes) repeating the service pattern. WFQ is also a work-conserving queuing discipline and thus will immediately move on to the next class in the service sequence upon finding an empty class queue.

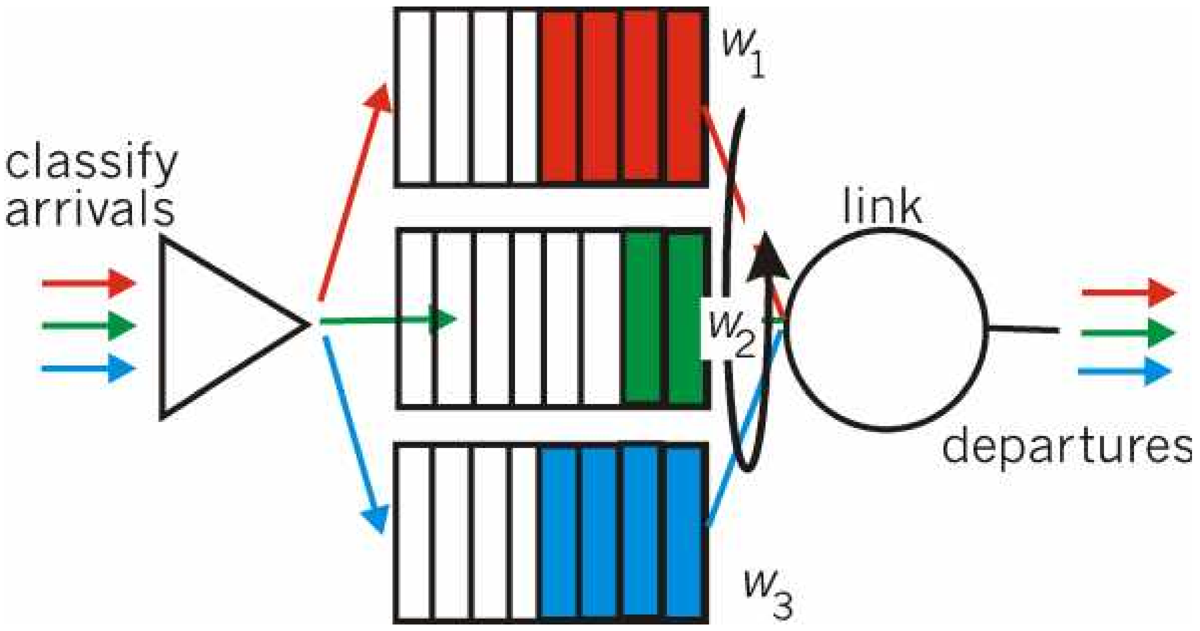


Figure 6.6-6: Weighted Fair Queuing (WFQ)

WFQ differs from round robin in that each class may receive a differential amount of service in any interval of time. Specifically, let each class, i, is assigned a weight, w_i. Under WFQ, during any interval of time during which there are class i packets to send, class i will then be guaranteed to receive a fraction of service equal to $w_i/(\quad w_j)$, where the sum in the denominator is taken over all classes that also have packets queued for transmission. In the worst case, even if all classes have queued packets, class i will still be guaranteed to receive a fraction $w_i/(\quad w_j)$, of the bandwidth.Thus, for a link with transmission rate R, class i will always achieve a throughput of at least $R{\cdot}w_i/(\quad w_j)$. Our description of WFQ has been an idealized one, as we have not considered the fact that packets are discrete units of data and a packet's transmission will not be interrupted to begin transmission another packet.