Unit-4

19ECS-232

COMPUTER NETWORKS

B.SRIDEVI

ASSISTANT PROFESSOR

DEPARTMENT OF CSE

**SYLLABUS**

**The Network Layer:** Introduction, Virtual Circuit and Datagram Networks, Inside Router, The Internet Protocol (IP), Routing Algorithms

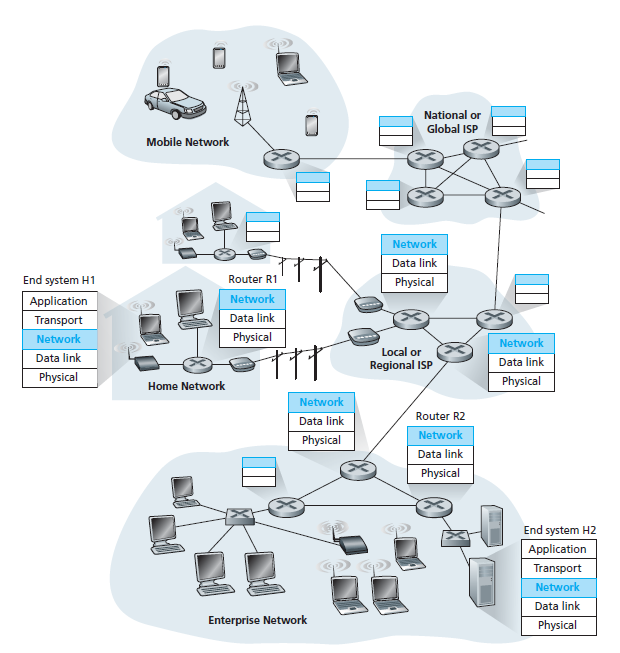
The transport layer provides various forms of process-to-process communication by relying on the network layer’s host-to-host communication service. Unlike the transport and application layers, there is a piece of the network layer in each and every host and router in the network. The network layer is also one of the most complex layers in the protocol stack.

**INTRODUCTION**

-The network layer in a host takes segments from the transport layer, encapsulates each segment into a datagram (that is, a network-layer packet), and then sends the datagrams to its nearby router.

-At the receiving host, the network layer receives the datagrams from its nearby router, extracts the transport-layer segments, and delivers the segments up to the transport layer.

* The primary role of the routers is to forward datagrams from input links to output links.
* In routers they are no upper layers above the network layer, because routers do not run application and transport-layer protocols.



**Fig:** The network layer

* **Forwarding and Routing**

-The role of the network layer is simple which is to move packets from a sending host to a receiving host.

-Two important network-layer functions can be identified as:

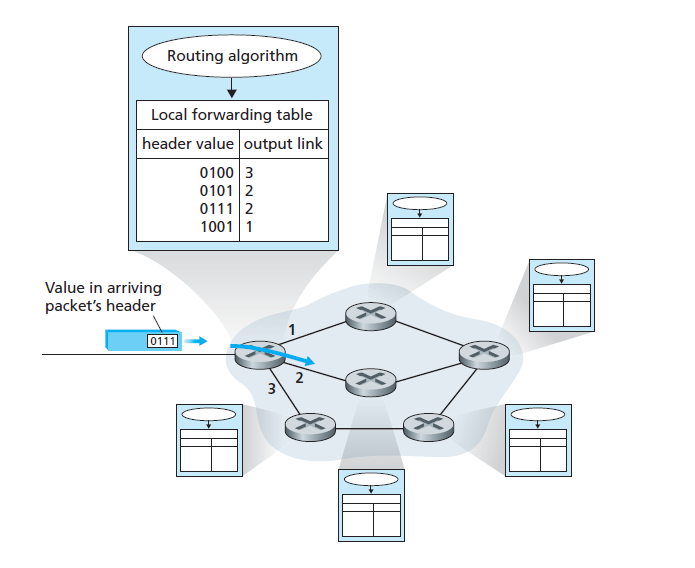
* **Forwarding:** When a packet arrives at a router’s input link, the router must move the packet to the appropriate output link.
* **Routing:** The network layer must determine the route or path taken by packets as they flow from a sender to a receiver.
* The algorithms that calculate these paths are referred to as **routing algorithms**.

-Forwarding refers to the router-local action of transferring a packet from an input link interface to the appropriate output link interface.

-Routing refers to the network-wide process that determines the end-to-end paths that packets take from source to destination.

-Every router has a **forwarding table**.

* A router forwards a packet by examining the value of a field in the arriving packet’s header, and then using this header value to index into the router’s forwarding table.
* The value stored in the forwarding table entry for that header indicates the router’s outgoing link interface to which that packet is to be forwarded.
* Depending on the network-layer protocol, the header value could be the destination address of the packet or an indication of the connection to which the packet belongs.

****

**Fig:** Routing algorithms determine values in forwarding tables

**Connection Setup:**

Some network-layer architectures require the routers along the chosen path from source to destination to handshake with each other in order to set up state before network-layer data packets within a given source-to-destination connection can begin to flow.

* **Network Service Models**

**-**In the sending host, when the transport layer passes a packet to the network layer, specific services that could be provided by the network layer include:

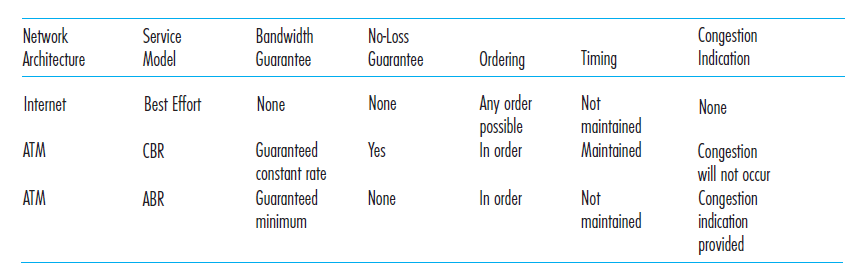
* + **Guaranteed Delivery:** This service guarantees that the packet will eventually arrive at its destination.
  + **Guaranteed Delivery with Bounded Delay:** This service not only guarantees delivery of the packet, but delivery within a specified host-to-host delay bound.

-The following services could be provided to a flow of packets between a given source and destination:

* **In-order Packet Delivery:** This service guarantees that packets arrive at the destination in the order that they were sent.
* **Guaranteed Minimal Bandwidth:** This network-layer service emulates the behavior of a transmission link of a specified bit rate between sending and receiving hosts.
* **Guaranteed Maximum Jitter:** This service guarantees that the amount of time between the transmission of two successive packets at the sender is equal to the

amount of time between their receipt at the destination.

* **Security Services:** Using a secret session key known only by a source and destination host, the network layer in the source host could encrypt the payloads of all datagrams being sent to the destination host.
* The network layer in the destination host would then be responsible for decrypting the payloads.
* The network layer could provide data integrity and source authentication services.



**Table:** Internet, ATM CBR, and ATM ABR service models

-The Internet’s network layer provides a single service, known as **best-effort service**.

-Other network architectures have defined and implemented service models thatgo beyond the Internet’s best-effort service.

* **For example:** The ATM network architecture provides for multiple service models, meaning that different connections can be provided with different classes of service within the same network.

-Two of the more important ATMservice models are constant bit rate and available bit rate service:

* **Constant bit rate (CBR) ATM network service:**
* The goal of CBR service is conceptually simple which is to provide a flow of packets with a virtual pipe whose properties are the same as if a dedicated fixed-bandwidth transmission link existed between sending and receiving hosts.
* **Available bit rate (ABR) ATM network service:**
* With the Internet offering so-called best-effort service, ATM’s ABR might best be characterized as being a slightly better-than-best-effort service.
* ATM ABR service can provide feedback to the sender in terms of a congestion notification bit, or an explicit rate at which to send.

**VIRTUAL CIRCUIT AND DATAGRAM NETWORKS**

-A network layer can provide connectionless service or connection service between two hosts.

-Network-layer connection and connectionless services in many ways parallel to transport-layer connection-oriented and connectionless services, but there are crucial differences:

* In the network layer, these services are host-to-host services provided by the network layer for the transport layer.
* In the transport layer, these services are process-to-process services provided by the transport layer for the application layer.

-Computer networks that provide only a connection service at the network layer are called **virtual-circuit** (**VC**) **networks** and computer networks that provide only a connectionless service at the network layer are called **datagram networks**.

* **Virtual-Circuit Networks**

**-**The network-layer connections are called **virtual circuits (VCs)**.

-A VC consists of

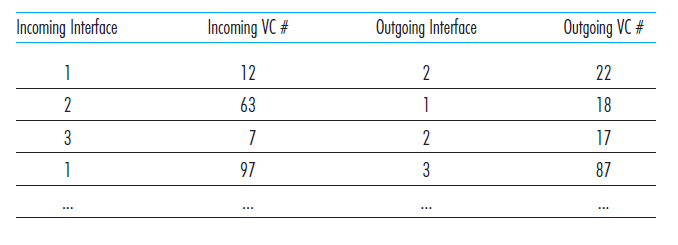
* A path (that is, a series of links and routers) between the source and destination hosts.
* VC numbers, one number for each link along the path.
* Entries in the forwarding table in each router along the path.

-A packet belonging to a virtual circuit will carry a VC number in its header.

-Because a virtual circuit may have a different VC number on each link, each intervening router must replace the VC number of each traversing packet with a new VC number.

-The new VC number is obtained from the forwarding table.

Whenever a new VC is established across a router, an entry is added to the forwarding table. Similarly, whenever a VC terminates, the appropriate entries in each table along its path are removed.

****

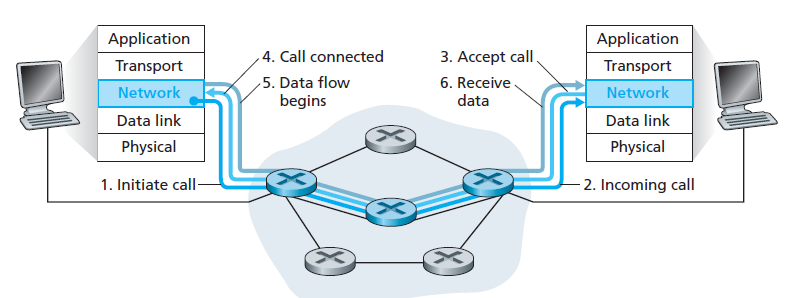
-In a VC network, the network’s routers must maintain **connection state information** for the ongoing connections.

* Specifically, each time a new connection is established across a router, a new connection entry must be added to the router’s forwarding table.
* Each time a connection is released, an entry must be removed from the table.

-There are three identifiable phases in a virtual circuit:

* **VC Setup:**
* During the setup phase, the sending transport layer contacts the network layer, specifies the receiver’s address, and waits for the network to set up the VC.
* The network layer determines the path between sender and receiver, that is, the series of links and routers through which all packets of the VC will travel.
* The network layer adds an entry in the forwarding table in each router along the path.
* During VC setup, the network layer may also reserve resources along the path of the VC.
* **Data Transfer:**
* Once the VC has been established, packets can begin to flow along with the VC.
* **VC Teardown:**
* This is initiated when the sender (or receiver) informs the network layer of its desire to terminate the VC.
* The network layer will then typically inform the end system on the other side of the network of the call termination and update the forwarding tables in each of the packet routers on the path to indicate that the VC no longer exists.

The messages that the end systems send into the network to initiate or terminate a VC, and the messages passed between the routers to set up the VC are known as **signaling messages**, and the protocols used to exchange these messages are often referred to as **signaling protocols.**



**Fig:** Virtual-Circuit Setup

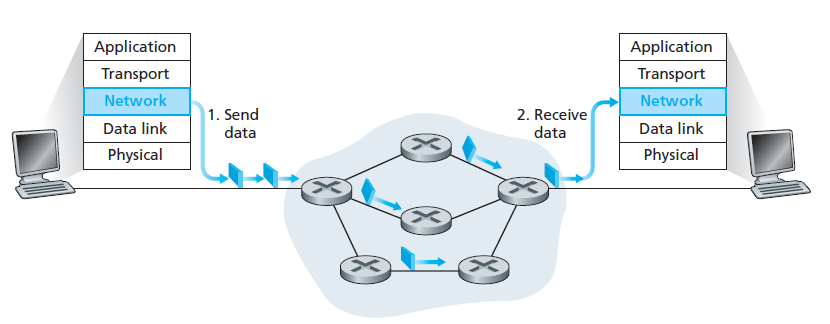
* **Datagram Networks**

-In a **datagram network**, each time an end system wants to send a packet, it stamps the packet with the address of the destination end system and then pops the packet into the network.

* + As a packet is transmitted from source to destination, it passes through a series

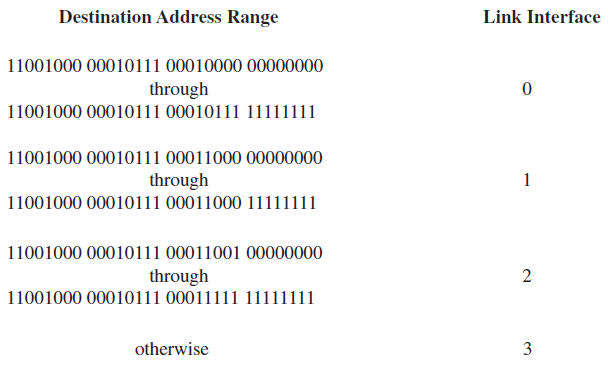
of routers.

* + Each router uses the packet’s destination address to forward the packet and has a forwarding table that maps destination addresses to link interfaces.
  + When a packet arrives at the router, the router uses the packet’s destination address to look up the appropriate output link interface in the forwarding table.

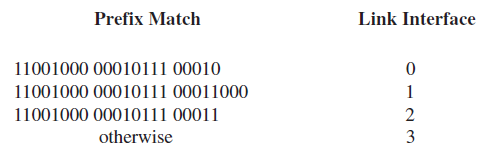


**Fig:** Datagram Network

-To get some further insight into the lookup operation in the router suppose that our router has four links, numbered 0 through 3, and that packets are to be forwarded to the link interfaces as follows:



-The following forwarding table with just four entries:



-With this style of forwarding table, the router matches a **prefix** of the packet’s destination

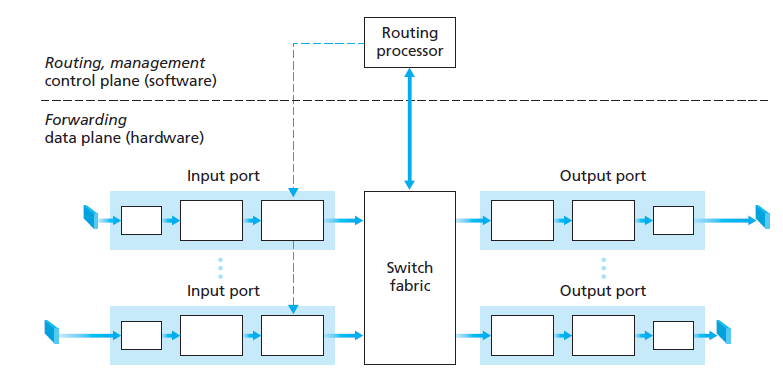
address with the entries in the table:

* If there’s a match, the router forwards the packet to a link associated with the match.
* When there are multiple matches, the router uses the **longest prefix matching rule**; that is, it finds the longest matching entry in the table and forwards the packet to the link interface associated with the longest prefix match.

-Forwarding tables in datagram networks can be modified at any time, a series of packets sent from one end system to another may follow different paths through the network and may arrive out of order.

**WHAT’S INSIDE A ROUTER?**

A high-level view of a generic router architecture is

****

**Fig:** Router Architecture

-Four router components can be identified:

* **Input ports:**
* An input port performs several key functions.
* It performs the physical layer function of terminating an incoming physical link at a router
* In the leftmost box of the input port and the rightmost box of the output port which is shown in the above diagram.
* 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.
* 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.
* **Switching fabric:**
* The switching fabric connects the router’s input ports to its output ports.
* This switching fabric is completely contained within the router.
* **Output ports:**
* An output port stores packets received from the switching fabric and transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions.
* **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.

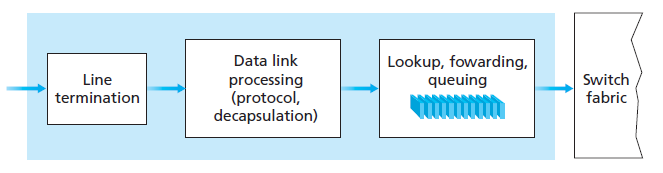
-A router’s input ports, output ports, and switching fabric together implement the forwarding function and are almost always implemented in hardware.

* These forwarding functions are sometimes collectively referred to as the **router forwarding plane**.

-A router’s control functions executing the routing protocols, responding to attached links that go up or down, and performing management functions.

* These **router control plane** functions are usually implemented in software and execute on the routing processor.
* **Input Processing**

-A more detailed view of input processing is



**Fig:** Input Port Processing

-The input port’s line termination function and link-layer processing implement the physical and link layers for that individual input link.

-The lookup performed in the input port is central to the router’s operation, it is here that the router uses the forwarding table to look up the output port to which an arriving packet will be forwarded via the switching fabric.

-The forwarding table is computed and updated by the routing processor, with a shadow copy typically stored at each input port.

-The forwarding table is copied from the routing processor to the line cards over a separate bus (e.g., a PCI bus) indicated by the dashed line from the routing processor to the input line cards.

-With a shadow copy, forwarding decisions can be made locally, at each input port, without invoking the centralized routing processor on a per-packet basis and thus avoiding a centralized processing bottleneck.

* **Switching**

-The switching fabric is at the very heart of a router, as it is through this fabric that the packets are actually switched (that is, forwarded) from an input port to an output port.

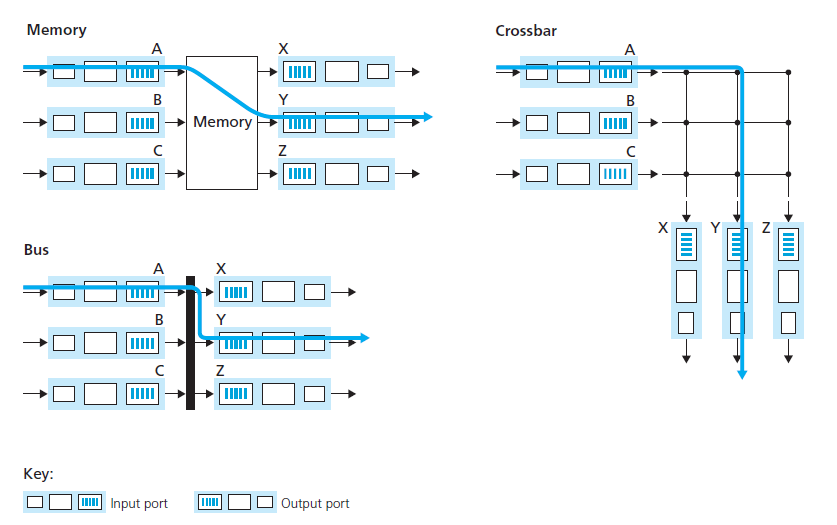
-Switching can be accomplished in a number of ways:

* **Switching via Memory:**
* The simplest, earliest routers were traditional computers, with switching between input and output ports being done under direct control of the CPU.
* Input and output ports functioned as traditional I/O devices in a traditional operating system.
* An input port with an arriving packet first signaled the routing processor.
* The packet was then copied from the input port into processor memory.
* The routing processor then extracted the destination address from the header, looked up the appropriate output port in the forwarding table, and copied the packet to the output port’s buffers.

Two packets cannot be forwarded at the same time, even if they have different destination ports, since only one memory read/write over the shared system bus can be done at a time.

* **Switching via a Bus:**
* In this approach, an input port transfers a packet directly to the output port over a shared bus, without intervention by the routing processor.
* This is typically done by having the input port pre-pend a switch-internal label (header) to the packet indicating the local output port to which this packet is being transferred.
* The packet is received by all output ports, but only the port that matches the label will keep the packet.
* If multiple packets arrive to the router at the same time, each at a different input port, all but one must wait since only one packet can cross the bus at a time.
* **Switching via an Interconnection Network:**
* One way to overcome the bandwidth limitation of a single, shared bus is to use a more sophisticated interconnection network.
* A crossbar switch is an interconnection network consisting of 2N buses that connect N input ports to N output ports.
* Each vertical bus intersects each horizontal bus at a cross-point, which can be opened or closed at any time by the switch fabric controller.
* When a packet arrives from port A and needs to be forwarded to port Y, the switch controller closes the cross-point at the intersection of busses A and Y, and port A then sends the packet onto its bus, which is picked up only by bus Y.
* A packet from port B can be forwarded to port X at the same time, since the A-to-Y and B-to-X packets use different input and output busses.

If two packets from two different input ports are destined to the same output port, then one will have to wait at the input, since only one packet can be sent over any given bus at a time.

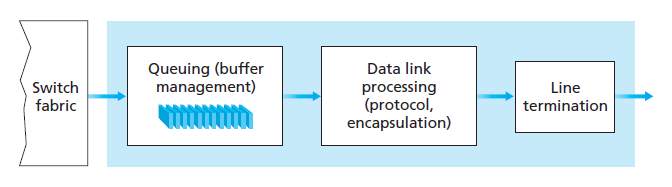


**Fig:** Three Switching Techniques

* **Output Processing**

-Output port processing, takes packets that have been stored in the output port’s memory and transmits them over the output link.

-This includes selecting and de-queueing packets for transmission, and performing the needed link-layer and physical-layer transmission functions.



**Fig:** Output Port Processing

* **Where Does Queueing Occur?**

-Packet queues may form at both the input ports and the output ports.

-The location and extent of queueing will depend on the traffic load, the relative speed of the switching fabric, and the line speed.

-These queues grow large, the router’s memory can eventually be exhausted and **packet loss** will occur when no memory is available to store arriving packets.

**Output Port Queuing**

-From the below figure: At time t, a packet has arrived at each of the incoming input ports, each destined for the uppermost outgoing port.

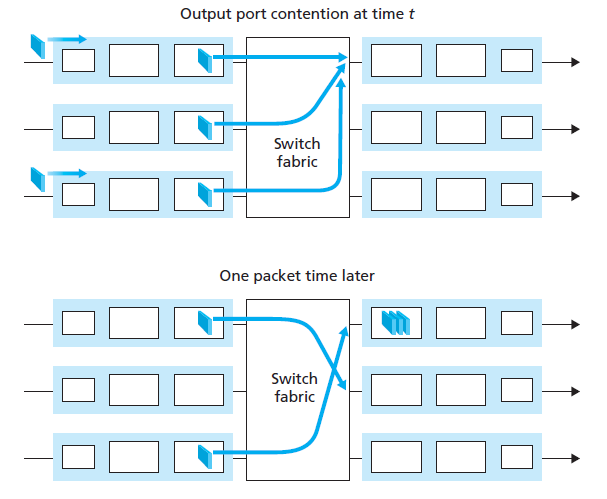
-Assuming identical line speeds and a switch operating at three times the line speed, one time unit later, all three original packets have been transferred to the outgoing port and are queued awaiting transmission.

-In the next time unit, one of these three packets will have been transmitted over the outgoing link.

-In our example, two new packets have arrived at the incoming side of the switch;

* One of these packets is destined for this uppermost output port.
* Router buffers are needed to absorb the fluctuations in traffic load.
* The amount of buffering needed is





**Fig:** Output port queuing

-A consequence of output port queuing is that a **packet scheduler** at the output port must choose one packet among those queued for transmission.

* This selection might be done on a simple basis, such as first-come-first-served (FCFS) scheduling.
* Packet scheduling plays a crucial role in providing **quality-of-service guarantees**.

-Similarly, if there is not enough memory to buffer an incoming packet, a decision must be made to either drop the arriving packet or remove one or more already-queued packets to make room for the newly arrived packet.

* In some cases, it may be advantageous to drop a packet before the buffer is full in order to provide a congestion signal to the sender.
* A number of packet-dropping and -marking policies, which collectively have become known as **active queue management** (**AQM**) algorithms.
* One of the most widely studied and implemented AQM algorithms is the **Random Early Detection** (**RED**) algorithm.
* Under RED, a weighted average is maintained for the length of the output queue.
* If the average queue length is less than a minimum threshold, then when a packet arrives, the packet is admitted to the queue.
* Conversely, if the queue is full or the average queue length is greater than a maximum threshold, then when a packet arrives, the packet is marked or dropped.

**Input Port Queuing**

-If the switch fabric is not fast enough that is relative to the input line speeds to transfer all arriving packets through the fabric without delay, then packet queuing can also occur at the input ports.

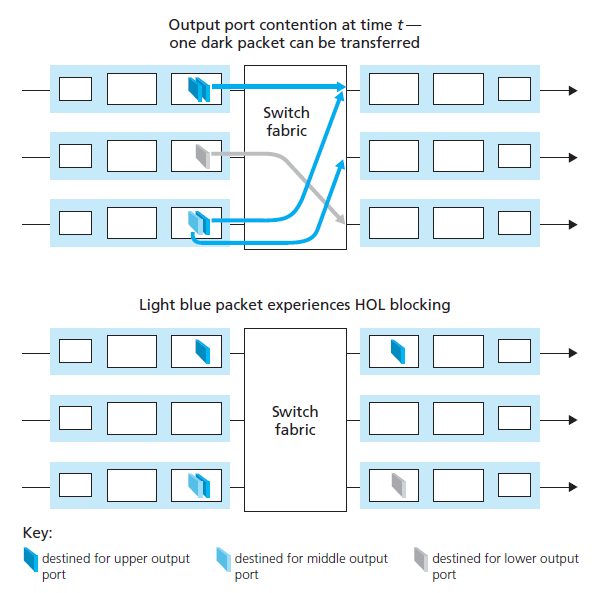
-Consider a crossbar switching fabric and suppose that

* All link speeds are identical
* That one packet can be transferred from any one input port to a given output port in the same amount of time it takes for a packet to be received on an input link.
* Packets are moved from a given input queue to their desired output queue in an FCFS manner.
* Multiple packets can be transferred in parallel, as long as their output ports are different.

-However, if two packets at the front of two input queues are destined for the same output queue, then one of the packets will be blocked and must wait at the input queue because the switching fabric can transfer only one packet to a given output port at a time.

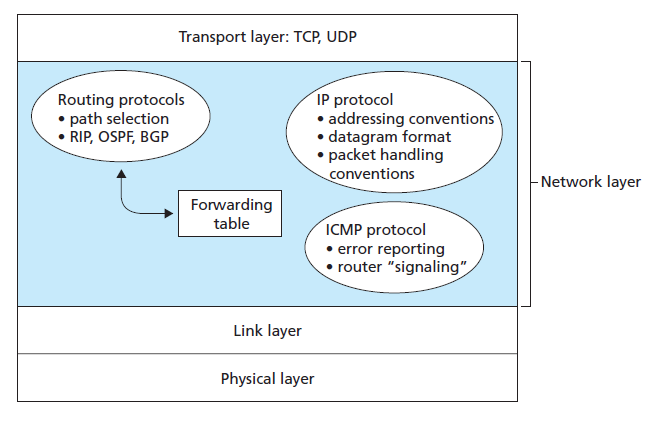
* If a packet which is blocked has a packet back of it then that packet must also have to wait.
* This phenomenon is known as **head-of-the-line** (**HOL**) **blocking:**

A queued packet in an input queue must wait for transfer through the fabric even though its output port is free because it is blocked by another packet at the head of the line.



**Fig:** HOL blocking at an input queued switch

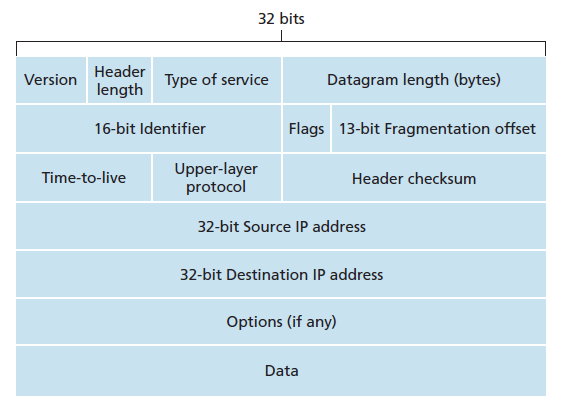
**THE INTERNET PROTOCOL (IP): FORWARDING AND ADDRESSING IN THE INTERNET**

****

**Fig:** A look inside the Internet’s network layer

-The Internet’s network layer has three major components.

* The first component is the IP protocol.
* The second major component is the routing component, which determines the path a datagram from source to destination.
* The final component of the network layer is a facility to report errors in datagrams and respond to requests for certain network-layer information.
* **Datagram Format**



**Fig:** IPv4 Datagram Format

-The key fields in the IPv4 datagram are the following:

* **Version Number:**
* Version number is of 4 bits.
* These bits specify the IP protocol version of the datagram.
* The router can determine how to interpret the remainder of the IP datagram as different versions of IP use different datagram formats.
* In the case of IPv4, the value of its four bits is set to 0100 which indicates 4.
* **Header Length:**
* An IPv4 datagram can contain a variable number of options which are included in the IPv4 datagram header.
* These 4 bits are needed to determine where in the IP datagram the data actually begins.
* Most IP datagrams do not contain options, so the typical IP datagram has a 20-byte header.
* **Type of Service:**
* It is 8-bit field.
* It is used to tell the network how to treat the IP packet.
* These bits are generally used to indicate the Quality of Service (QoS) for the IP Packet.
* The type of service (TOS) bits were included in the IPv4 header to allow different types of IP datagrams to be distinguished from each other.
* **Example:** datagrams particularly requiring low delay, high throughput, or reliability.
* **Datagram Length:**
* This is the total length of the IP datagram (header plus data), measured in bytes.
* Since this field is 16 bits long, theoretical maximum size of the IP datagram is 65,535 bytes.
* **Time-to-Live:**
* Time to live (or TTL) is an 8-bit field.
* It indicates the maximum time the datagram will be live in the internet system.
* The time here is measured in seconds and in case the value of TTL is zero, the datagram is dropped.
* Every time a datagram is processed, it’s Time to live is decreased by one second.
* TTL can be between 0 – 255.
* The main purpose of TTL is to prevent the IP datagram’s from looping around forever.
* **Protocol:**
* This field is used only when an IP datagram reaches its final destination.
* The value of this field indicates the specific transport-layer protocol to which the data portion of this IP datagram should be passed.
* The protocol number is the glue that binds the network and transport layers together, whereas the port number is the glue that binds the transport and application layers together.
* **Header Checksum:**
* The header checksum aids a router in detecting bit errors in a received IP datagram.
* The header checksum is computed by treating each 2 bytes in the header as a number and summing these numbers using 1s complement arithmetic.
* **Source Address:**
* Source IP Address is a 32-bit field.
* It contains the logical address of the sender of the datagram.
* **Destination Address:**
* Destination IP Address is a 32-bit field.
* It contains the logical address of the receiver of the datagram.
* **Options:**
* The options fields allow an IP header to be extended.
* Header options were meant to be used rarely to save overhead by not including the information in options fields in every datagram header.
* Some datagrams may require options processing and others may not, the amount of time needed to process an IP datagram at a router can vary greatly.
* These considerations become particularly important for IP processing in high-performance routers and hosts.
* **Data (payload):**
* The last and most important field is the data field of the IP datagram contains the transport-layer segment to be delivered to the destination.
* The data field can carry other types of data, such as ICMP messages.

**IP Datagram Fragmentation**

-The data in the IP datagram is divided into two or more smaller IP datagrams, encapsulate each of these smaller IP datagrams in a separate link-layer frame; and send these frames over the outgoing link.

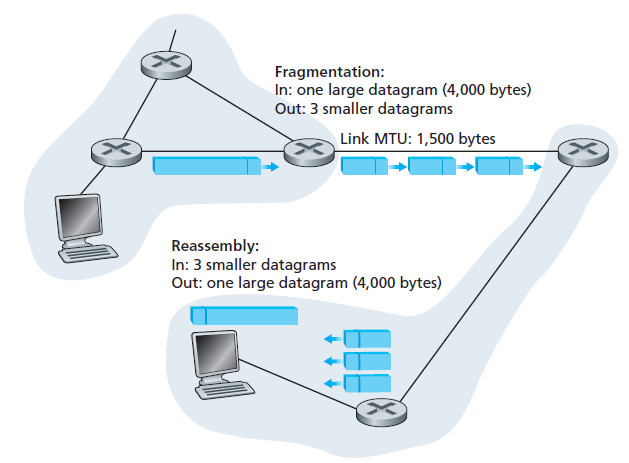
-Each of these smaller datagrams is referred to as a **fragment**.

-Fragments need to be reassembled before they reach the transport layer at the destination.

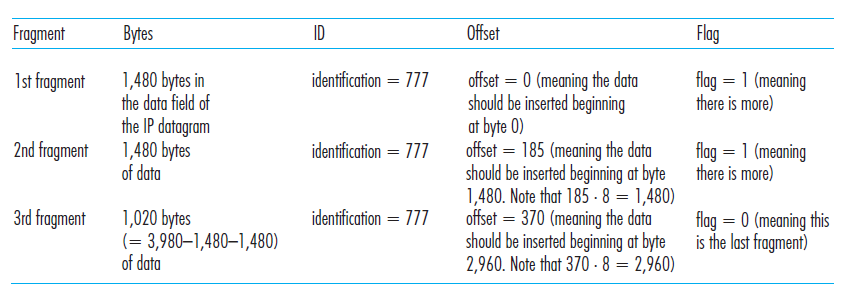
-The destination host to perform these reassembly tasks, put identification, flag, and fragmentation offset fields in the IP datagram header.

* **Identification:**
* Identification is 16-bit field.
* The Identification field is needed to allow the destination host to determine which packet a newly arrived fragment belongs to.
* All the fragments of a packet contain the same Identification value.
* **Fragment offset:**
* Fragment Offset is a 13-bit field.
* The Fragment offset tells where in the current packet this fragment belongs i.e. it indicates the position of a fragmented datagram in the original unfragmented IP datagram.
* **Flag:**
* In order for the destination host to be absolutely sure it has received the last fragment of the original datagram, the last fragment has a flag bit set to 0, whereas all the other fragments have this flag bit set to 1.

At the destination, the payload of the datagram is passed to the transport layer only after the IP layer has fully reconstructed the original IP datagram. If one or more of the fragments does not arrive at the destination, the incomplete datagram is discarded and not passed to the transport layer.



**Fig:** IP fragmentation and reassembly



**Table:**  IP fragments

* **IPv4 Addressing**

-A host typically has only a single link into the network.

-When IP in the host wants to send a datagram, it does so over the link.

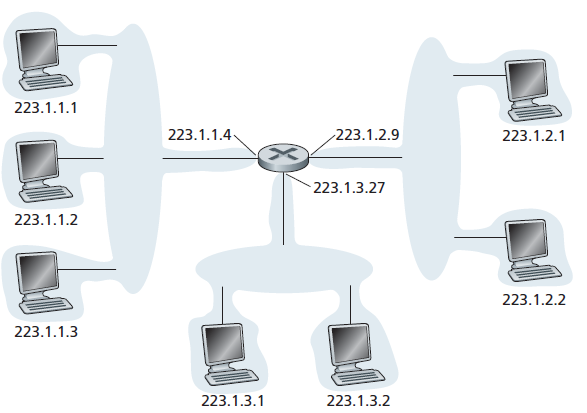
-The boundary between the host and the physical link is called an **interface**.

-A router’s job is to receive a datagram on one link and forward the datagram on some other link, a router necessarily has two or more links to which it is connected.

-A router thus has multiple interfaces, one for each of its links.

-As every host and router is capable of sending and receiving IP datagrams, IP requires each host and router interface to have its own IP address.

* + Thus, an IP address is technically associated with an interface, rather than with the host or router.



**Fig:** Interface addresses and subnets

-Each IP address is 32 bits long (equivalently, 4 bytes), and there are thus a total of 2^32 possible IP addresses.

-These addresses are typically written in so-called **dotted-decimal notation**, in which each byte of the address is written in its decimal form and is separated by a period (dot) from other bytes in the address.

* + For example: Consider the IP address 193.32.216.9.
  + The address 193.32.216.9 in binary notation is

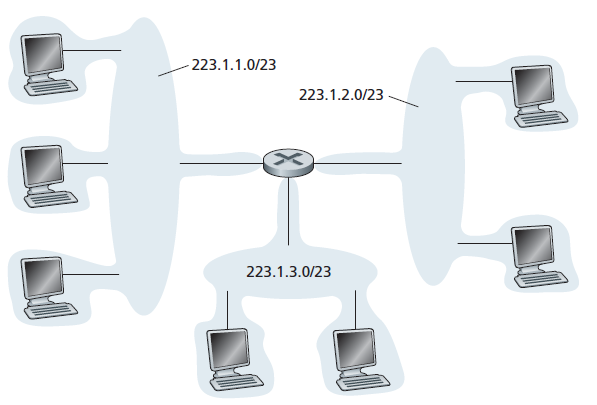
11000001 00100000 11011000 00001001

-Each interface on every host and router in the global Internet must have an IP address that is globally unique.

-A portion of an interface’s IP address will be determined by the subnet to which it is connected.

-In IP terms, this network interconnecting three host interfaces and one router interface forms a **subnet.**

-IP addressing assigns an address to this subnet: 223.1.1.0/24, where the /24 notation, sometimes known as a **subnet mask**, indicates that the leftmost 24 bits of the 32-bit quantity define the subnet address.



**Fig:** Subnet Addresses

-The Internet’s address assignment strategy is known as **Classless Interdomain Routing** (**CIDR**).

-CIDR generalizes the notion ofsubnet addressing.

* The 32-bit IP address is divided into two parts and again has the dotted-decimal form a.b.c.d/x, where x indicates the number of bits in the first part of the address.
* The x most significant bits of an address of the form a.b.c.d/x constitute the network portion of the IP address, and are often referred to as the **prefix** of the address.
* An organization is typically assigned a block of contiguous addresses, that is, a range of addresses with a common prefix.
* The remaining bits then identify the specific hosts in the organization.
* The organization’s internal structure might be such that these rightmost bits are used for subnetting within the organization.

-Before CIDR was adopted, the network portions of an IP address were constrained to be 8, 16, or 24 bits in length, an addressing scheme known as **classful** **addressing**.

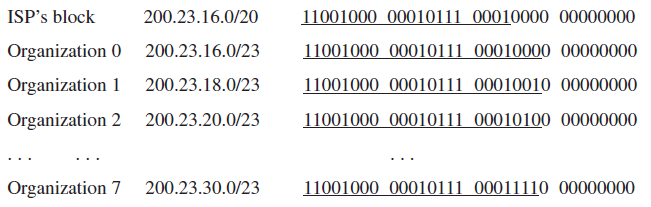
-Subnets with 8-, 16-, and 24-bit subnet addresses were known as class A, B, and C networks, respectively.

-The IP broadcast address 255.255.255.255.

-When a host sends a datagram with destination address 255.255.255.255, the message is delivered to all hosts on the same subnet.

* **Obtaining a Block of Addresses**

-In order to obtain a block of IP addresses for use within an organization’s subnet, a network administrator might first contact its ISP, which would provide addresses from a larger block of addresses that had already been allocated to the ISP.



-IP addresses are managed under the authority of the Internet Corporation for Assigned Names and Numbers (ICANN).

-The role of the non-profit ICANN organization is not only to allocate IP addresses, but also to manage the DNS root servers.

-It also has the very contentious job of assigning domain names and resolving domain name disputes.

-The ICANN allocates addresses to regional Internet registries and handle the allocation/management of addresses within their regions.

* **Dynamic Host Configuration Protocol**

**-**Every computer on a network requires IP address for communication.

-DHCP is a network management protocol used to dynamically assign an Internet Protocol

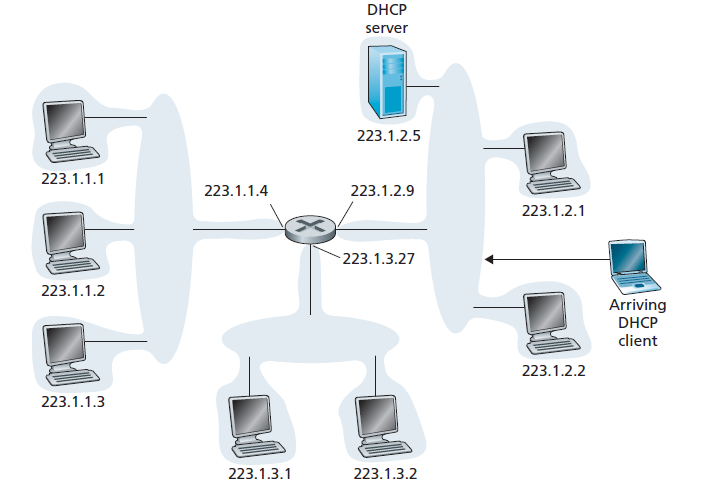
(IP) address to any device, or node, on a network so they can communicate.

-DHCP automatically and centrally manages these configurations rather than requiring

network administrators to manually assign IP addresses to all network devices.

-In addition to host IP address assignment, DHCP also allows a host to learn additional information, such as its subnet mask, the address of its first-hop router (often called the default gateway), and the address of its local DNS server.

-Because of DHCP’s ability to automate the network-related aspects of connecting a host into a network, it is often referred to as a **plug-and-play protocol**.

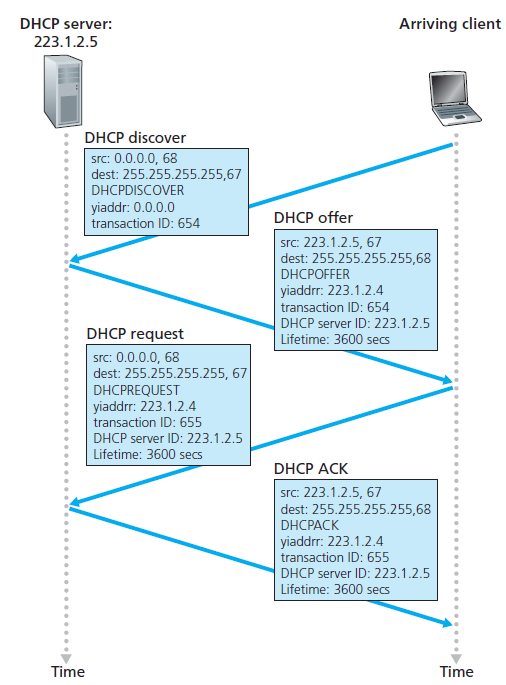


**Fig:** DHCP client-server scenario

-For a newly arriving host, the DHCP protocol is a four-step process

* + **DHCP server discovery:**
  + The first task of a newly arriving host is to find a DHCP server with which to interact.
  + This is done using a **DHCP discover message**, which a client sends within a UDP packet to port 67.
  + The DHCP client creates an IP datagram containing its DHCP discover message along with the broadcast destination IP address of 255.255.255.255 and a “this host” source IP address of 0.0.0.0.
  + The DHCP client passes the IP datagram to the link layer, which then broadcasts this frame to all nodes attached to the subnet.
  + **DHCP server offer(s):**
* A DHCP server receiving a DHCP discover message responds to the client with a **DHCP offer message** that is broadcast to all nodes on the subnet, again using the IP broadcast address of 255.255.255.255.
* Since several DHCP servers can be present on the subnet, the client may find itself in the position of being able to choose from among several offers.
* Each server offer message contains the transaction ID of the received discover message, the proposed IP address for the client, the network mask, and an IP **address lease time.**
  + **DHCP request:**
* The newly arriving client will choose from among one or more server offers and respond to its selected offer with a **DHCP request message**, echoing back the configuration parameters.
  + **DHCP ACK:**
* The server responds to the DHCP request message with a **DHCP ACK message**, confirming the requested parameters.

Once the client receives the DHCP ACK, the interaction is complete and the client can use the DHCP-allocated IP address for the lease duration.

****

**Fig:** DHCP client-server interaction

* **Network Address Translation (NAT)**

-The idea of NAT is to allow multiple devices to access the Internet through a single public address.

-To access the Internet, one public IP address is needed, but we can use a private IP address in our private network.

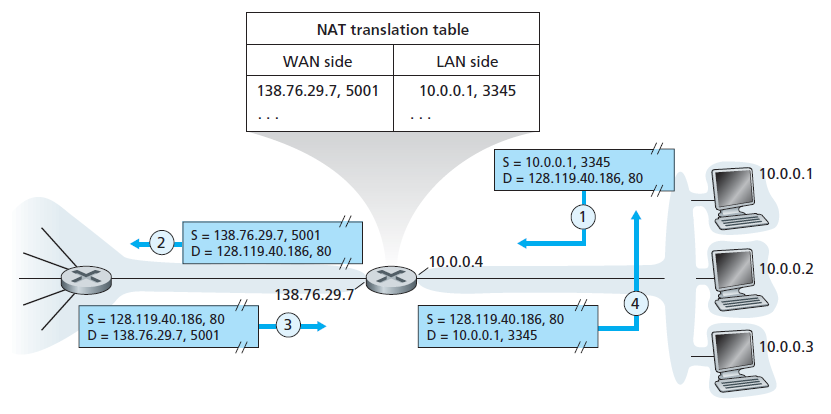
-To achieve this, the translation of private IP address to a public IP address is required.

-**Network Address Translation (NAT)** is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts.

-The outside world does not know about the private IP addresses.

-So, sending and receiving of the packets is done through the public address only.

* + If a host in the private network wants to send a packet to the other network, then it sends the packet with the private address to the router.
  + The router maintains a NAT translation table.
  + Where it makes the corresponding entries of IP address and port number in the NAT table.
  + It replaces the private address with the public address.



**Fig:** Network Address Translation

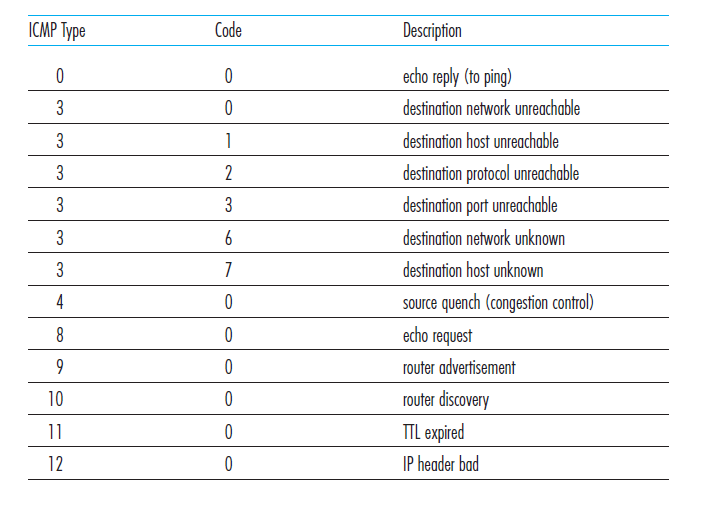
* **Internet Control Message Protocol (ICMP)**

**-**ICMP, is used by hosts and routers to communicate network-layer information to each other.

-The most typical use of ICMP is for error reporting.

-ICMP is often considered part of IP but architecturally it lies just above IP, as ICMP messages are carried inside IP datagrams.

-ICMP messages have a type and a code field, and contain the header and the first 8 bytes of the IP datagram that caused the ICMP message to be generated in the first place.



**Fig:** ICMP message types

-Interesting ICMP message is the source quench message.

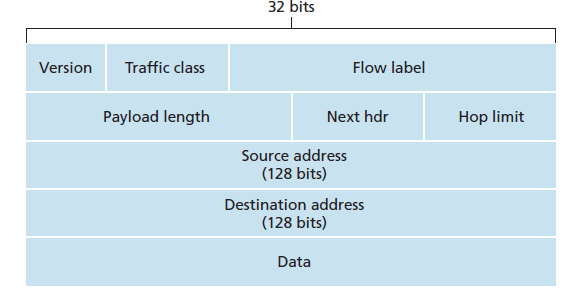
* Its original purpose was to perform congestion control to allow a congested router to send an ICMP source quench message to a host to force that host to reduce its transmission rate

-The TTL expired message is sent when a packet is dropped because it’s TtL (Time to live) counter has reached zero.

* One use of this error message is the **trace route.**
* Trace route finds the routers along the path from the host to a destination.
* The method is simply to send a sequence of packets to the destination, first with a TtL of 1, then a TtL of 2, 3, and so on.
* The counters on these packets will reach zero at successive routers along the path.
* These routers will each obediently send a TTL expired (type 11 code 0) message back to the host.
* From those messages, the host can determine the IP addresses of the routers along the path, as well as keep statistics and timings on parts of the path.
* **IPv6 Datagram Format**

-The most important changes introduced in IPv6 are evident in the datagram format:

* + **Expanded addressing capabilities:**
  + IPv6 increases the size of the IP address from 32 to 128 bits.
  + This ensures that the world won’t run out of IP addresses.
  + In addition to unicast and multicast addresses, IPv6 has introduced a new type of address, called an **anycast address**, which allows a datagram to be delivered to any one of a group of hosts.
  + **A streamlined 40-byte header:**
* A number of IPv4 fields have been dropped or made optional.
* The resulting 40-byte fixed-length header allows for faster processing of the IP datagram.
  + **Flow labeling and priority:**
* Labeling of packets belonging to particular flows for which the sender requests special handling, such as a nondefault quality of service or real-time service.



**Fig:** IPv6 datagram format

-The following fields are defined in IPv6:

* + **Version:**
* The first header field i.e. version is a 4 bit field that keeps track of which version of the protocol the datagram belongs to.
* In the case of IPv6, the value of its four bits is set to 0110 which indicates 6.
  + **Traffic Class:**
* The size of Traffic Class field is 8 bits.
* Traffic Class field is similar to the IPv4 Type of Service (ToS) field.
* This field indicates the IPv6 packet’s class or priority.
  + **Flow Label:**
* The size of Flow Label field is 20 bits.
* The purpose of Flow Label field is to indicate that the packet belongs to a specific sequence of packets between a source and destination and can be used to prioritized delivery of packets for services like voice.
  + **Payload length:**
* This 16-bit value is treated as an unsigned integer giving the number of bytes in the IPv6 datagram following the fixed-length, 40-byte datagram header.
  + **Next Header:**
* The size of the Next Header field is 8 bits.
* This field identifies the protocol to which the contents of this datagram will be delivered.
* The field uses the same values as the protocol field in the IPv4 header.
  + **Hop Limit:**
* The size of the Hop Limit field is 8 bits.
* The contents of this field are decremented by one by each router that forwards the datagram.
* If the hop limit count reaches zero, the datagram is discarded.
  + **Source Address:**
* The size of the source address field is 128 bits.
* This field indicates the address of originator of the packet.
  + **Destination Address:**
* The size of the destination address field is 128 bits.
* This field provides the address of intended recipient of the packet.
  + **Data:**
* This is the payload portion of the IPv6 datagram.
* When the datagram reaches its destination, the payload will be removed from the IP datagram and passed on to the protocol specified in the next header field.

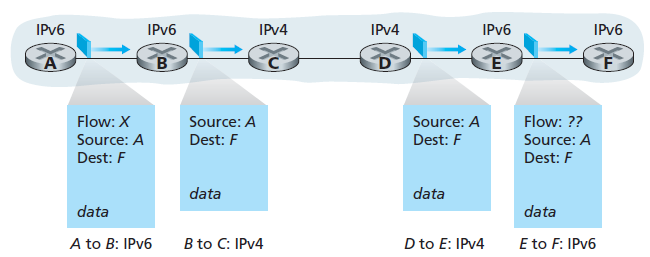
**Transitioning from IPv4 to IPv6**

-The most straightforward way to introduce IPv6 nodes is a **dual-stack** approach, where IPv6 nodes also have a complete IPv4 implementation.

-Such a node, referred to as an IPv6/IPv4 node, has the ability to send and receive both IPv4 and IPv6 datagrams.

-In the dual-stack approach, if either the sender or the receiver is only IPv4- capable, an IPv4 datagram must be used.

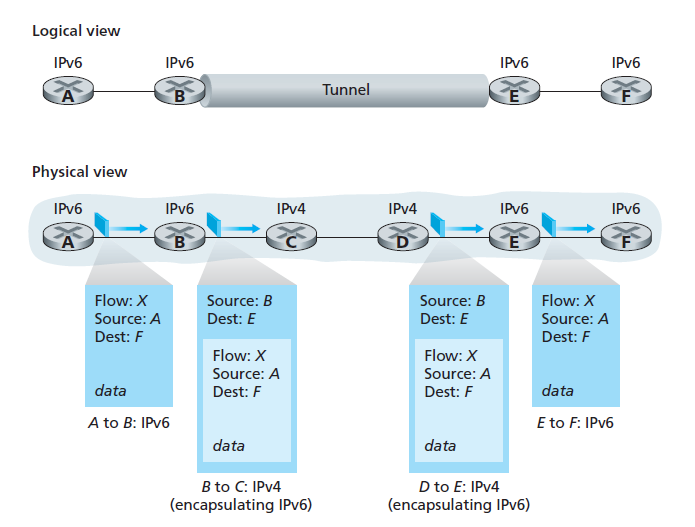
* + As a result, it is possible that two IPv6 capable nodes can end up, in essence, sending IPv4 datagrams to each other.



**Fig:** A dual-stack approach

-Consider Node A is IPv6-capable and wants to send an IP datagram to Node F*,* which is also IPv6-capable.

* + Nodes A and B can exchange an IPv6 datagram.
  + However, Node B must create an IPv4 datagram to send to C.
  + Certainly, the data field of the IPv6 datagram can be copied into the data field of the IPv4 datagram and appropriate address mapping can be done.
  + However, in performing the conversion from IPv6 to IPv4, there will be IPv6-specific fields in the IPv6 datagram (for example, the flow identifier field) that have no counterpart in IPv4.
  + The information in these fields will be lost.
  + Thus, even though E and F can exchange IPv6 datagrams, the arriving IPv4 datagrams at E from D.

****

**Fig:** Tunneling

-An alternative to the dual-stack approach, is known as **tunneling**.

-The basic idea behind tunneling is the following.

* Suppose two IPv6 nodes (for example, B and E in the above figure) want to interoperate using IPv6 datagrams but are connected to each other by intervening IPv4 routers.
* The intervening set of IPv4 routers between two IPv6 routers is considered as a **tunnel.**
* With tunneling, the IPv6 node on the sending side of the tunnel (for example, B) takes the entire IPv6 datagram and puts it in the data (payload) field of an IPv4 datagram.
* This IPv4 datagram is then addressed to the IPv6 node on the receiving side of the tunnel (for example, E) and sent to the first node in the tunnel (for example, C).
* The intervening IPv4 routers in the tunnel route this IPv4 datagram among themselves, just as they would any other datagram.
* The IPv6 node on the receiving side of the tunnel eventually receives the IPv4 datagram, determines that the IPv4 datagram contains an IPv6 datagram, extracts the IPv6 datagram, and then routes the IPv6 datagram exactly as it would if it had received the IPv6 datagram from a directly connected IPv6 neighbor.