Routing and Forwarding

- Figure 1 shows a simple network with two hosts, H1 and H2, and several routers on the path between H1 and H2.
- Suppose that H1 is sending information to H2, and consider the role of the network layer in these hosts and in the intervening routers.
- The network layer in H1 takes segments from the transport layer in H1, encapsulates each segment into a datagram (that is, a network-layer packet), and then sends the datagrams to its nearby router, R1.
- At the receiving host, H2, the network layer receives the datagrams from its nearby router R2, extracts the transport-layer segments, and delivers the segments up to the transport layer at H2.
- The primary role of the routers is to forward datagrams from input links to outputlinks.

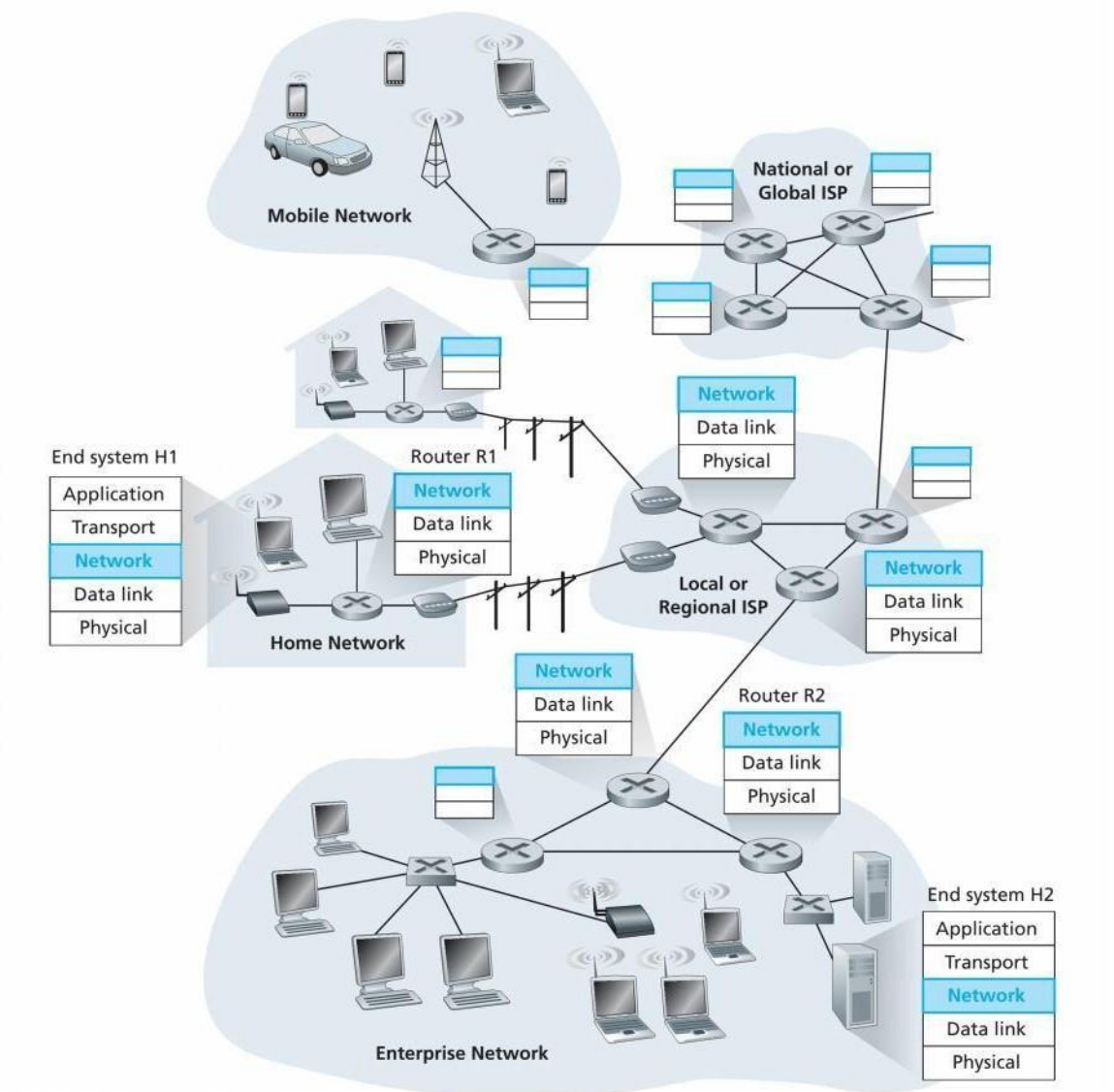


Fig. 1 Network Layer

- The role of the network layer is thus deceptively simple—to move packets from a sending host to a receiving host. To do so, two important network-layer functions can beidentified:
  - **Forwarding**: When a packet arrives at a router's input link, the router must move the packet to the appropriate output link.
  - For example, a packet arriving from Host H1 to Router R1 must be forwarded to the next router on a path to H2.
  - **Routing:** Routing is the process of selecting best paths in anetwork.
  - 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. A routing algorithm would determine, for example, the path along which packets flow from H1 to H2.
- 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 packetbelongs.

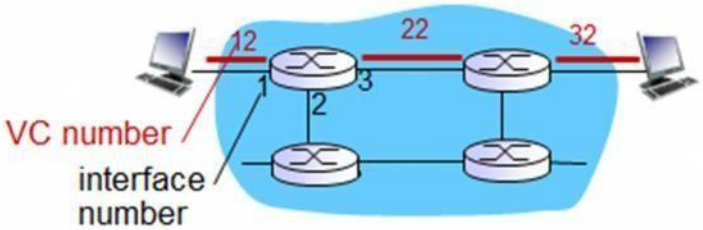
Network service model

- Services provided by network layer for individual datagrams
  1. **Guaranteed delivery:** This service guarantees that the packet will eventually arrive at its destination.
  2. **Guaranteed delivery with bounded delay:** This service not only guarantees delivery of the packet, but delivery within a specified host-to-host delay bound (for example, within 100 msec).
- Services provided by network layer for a flow of datagrams
  3. **In-order packet delivery:** This service guarantees that packets arrive at the destination in the order that they were sent.
  4. **Guaranteed minimal bandwidth:** This network-layer service emulates the behavior of a transmission link of a specified bit rate (for example, 1 Mbps) between sending and receiving hosts. As long as the sending host transmits bits at a rate below the specified bit rate, then no packet is lost.
  5. **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.
  6. **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. With such a service, confidentiality would be provided to all transport-layer segments (TCP and UDP) between the source and destination hosts.

Virtual and Datagram networks

Virtual Circuit Switching (Connection Oriented Service)

- A VC consists of
  1. a path (that is, a series of links and routers) between the source and destinationhosts
  2. VC numbers, one number for each link along the path
  3. 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.



forwarding table in northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

Fig. 2 A simple virtual circuit network

- The numbers next to the links of R1 in above figure are the link interfacenumbers.
- Suppose now that Host A requests that the network establishes a VC between itself and Host B.
- Suppose also that the network chooses the path A-R1-R2-B and assigns VC numbers 12, 22 and 32 to the three links in this path for this virtual circuit.
- In this case, when a packet in this VC leaves Host A, the value in the VC number field in the packet header is 12; when it leaves R1, the value is 22; and when it leaves R2, the value is32.
- How does the router determine the replacement VC number for a packet traversing the router? For a VC network, each router's forwarding table includes VC number translation; for example, the forwarding table in R1 might look something like above fig. 2
- Whenever a new VC is established across a router, an entry is added to the forwardingtable.
- Similarly, whenever a VC terminates, the appropriate entries in each table along its path are removed.
- A path from the source router to the destination router must be established before any data packets can be sent.
- This connection is called a VC (virtual circuit), and the subnet is called a virtual-circuit subnet.

- When a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside therouters.
- That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works.

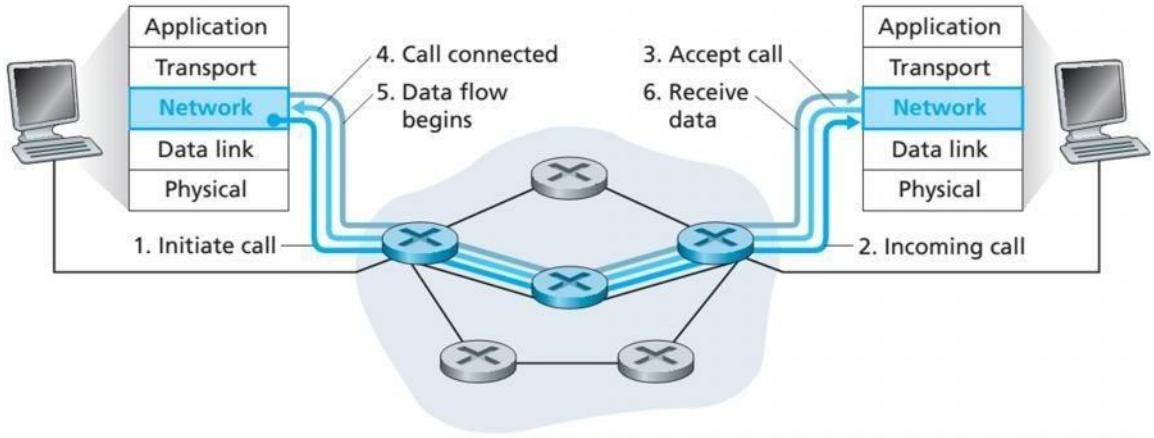


Fig. 3 Virtual-Circuit Setup

- There are three identifiable phases in a virtual circuit:
  1. **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 also determines the VC number for each link along the path.
     - Finally, 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 (for example, bandwidth) along the path of the VC.
  2. **Data transfer:**
     - As shown in Figure 3, once the VC has been established, packets can begin to flow along the VC.
  3. **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 longerexists.

Datagram Network (Connection-Less Service)

- In connection less service, packets are injected into the subnet individually and routed independently of each other.
- No advance setup is needed. In this context, the packets are frequently called datagrams (in analogy with telegrams) and the subnet is called a **datagram subnet**.
- Suppose that the process P1 in Figure 4 has a long message for P2. It hands the message to the transport layer with instructions to deliver it to process P2 on host H2.

- Let us assume that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4 and sends each of them in turn to router A using some point-to-point protocol, for example, PPP.

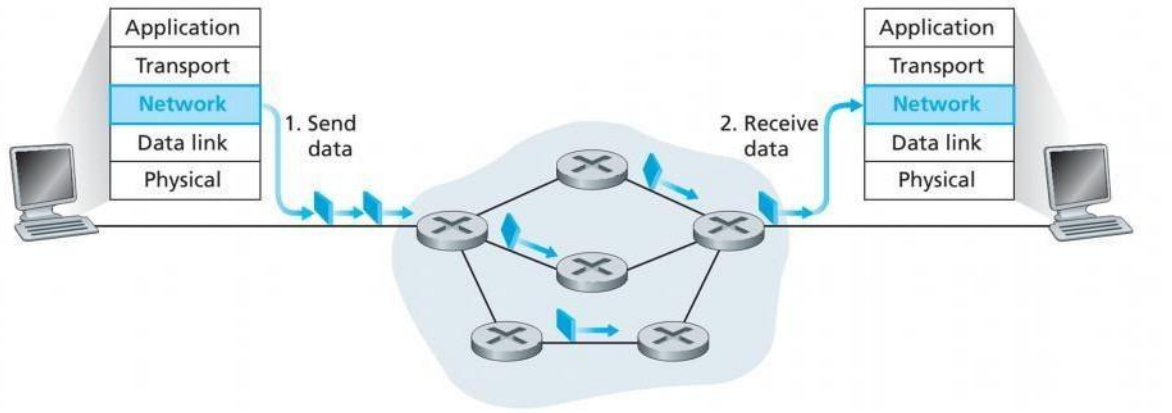


Fig. 4 Datagram Network

- At this point the carrier takes over. Every router has an internal table telling it where to send packets for each possible destination. Each table entry is a pair consisting of a destination and the outgoing line to use for that destination.
- Only directly-connected lines can be used.

Datagram Network vs. Virtual Circuit Network

Issue	Datagram	Virtual Circuit
Connection Setup	None	Required
Addressing	Packet contains full source and destination address	Packet contains short virtual circuit number identifier.
State Information	None other than router table containing destination network	Each virtual circuit number entered to table on setup, used for routing.
Routing	Packets routed independently	Route established at setup, all packets follow same route.
Effect of Router Failure	Only on packets lost during crash	All virtual circuits passing through failed router terminated.
Congestion Control	Difficult since all packets routed independently router resource requirements can vary.	Simple by pre-allocating enough buffers to each virtual circuit at setup, since maximum number of circuits fixed.

Router architecture

- Routers have four components:
    1. Input ports
    2. Switching fabric
    3. Output ports
    4. Routing processor

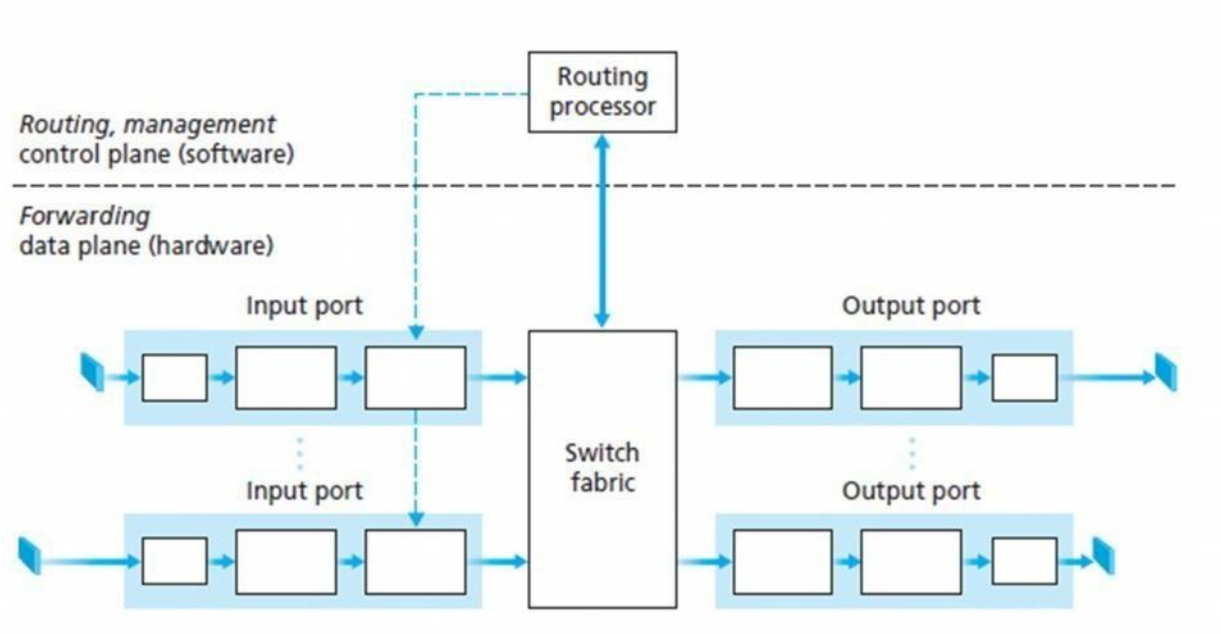


Fig. 5 Router architecture

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 5.
- 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 switchingfabric.
- 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 outputports.
- This switching fabric is completely contained within the router - a network inside of a network 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-layerfunctions.
- 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 linecard.

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.
- It also performs the network management functions.

Types of switching fabrics

- Three types of switching fabrics
    1. Switching via memory
    2. Switching via a bus
    3. Switching via an interconnection network

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 (routing processor).
- An input port with an arriving packet first signalled the routing processor via an interrupt. 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'sbuffers.
- In this scenario, if the memory bandwidth is such that B packets per second can be written into, or read from, memory, then the overall forwarding throughput must be less than B/2.
- Note also that 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.

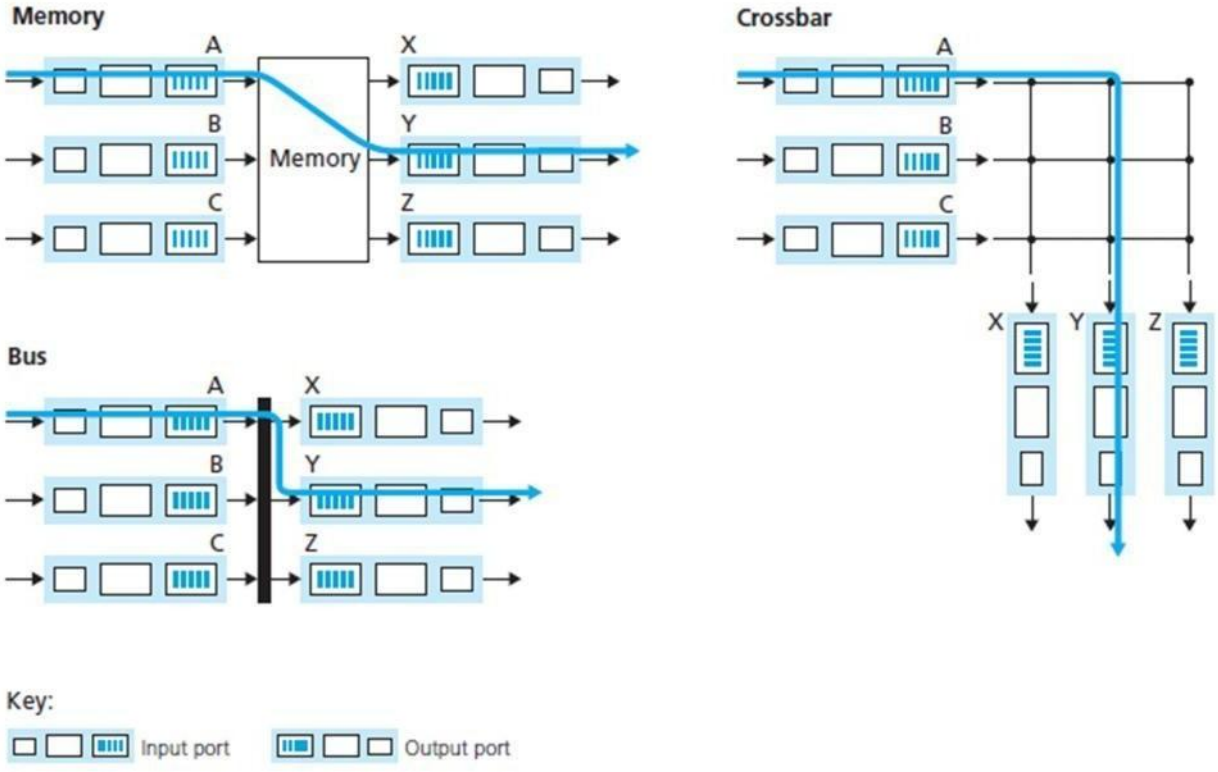


Fig. 6 Three switching techniques

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 and transmitting the packet onto the bus.
- The packet is received by all output ports, but only the port that matches the label will keep the packet.
- The label is then removed at the output port, as this label is only used within the switch to cross the bus.
- 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. Because every packet must cross the single bus, the switching speed of the router is limited to the busspeed.

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, such as those that have been used in the past to interconnect processors in a multiprocessor computer architecture.
- A crossbar switch is an interconnection network consisting of 2N buses that connect N input ports to N output ports, as shown in Figure 6.
- Each vertical bus intersects each horizontal bus at a crosspoint, which can be opened or closed at any time by the switch fabric controller (whose logic is part of the switching fabric itself).
- When a packet arrives from port A and needs to be forwarded to port Y, the switch controller closes the crosspoint 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.
- Note that 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.
- Thus, unlike the previous two switching approaches, crossbar networks are capable of forwarding multiple packets in parallel.
- However, 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 atime.

IPv4 datagram format

- **Version number**: These 4 bits specify the IP protocol version of the datagram. It determines how to interpret the header. Currently the only permitted values are 4 (0100) or 6(0110).
- **Header length:** Specifies the length of the IP header, in 32-bitwords.
- **Type of service:** The type of service (TOS) bits were included in the IPv4 header to allow different types of IP datagrams (for example, datagrams particularly requiring low delay, high throughput, or reliability) to be distinguished from each other.
- **Datagram length:** This is the total length of the IP datagram (header plus data), measured in bytes.
- **Identifier:** Uniquely identifies the datagram. It is incremented by 1 each time a datagram is sent. All fragments of a datagram contain the same identification value. This allows the destination host to determine which fragment belongs to which datagram.
- **Flags:** 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.
- **Fragmentation offset:** When fragmentation of a message occurs, this field specifies the offset, or position, in the overall message where the data in this fragment goes. It is specified in units of 8 bytes (64 bits).

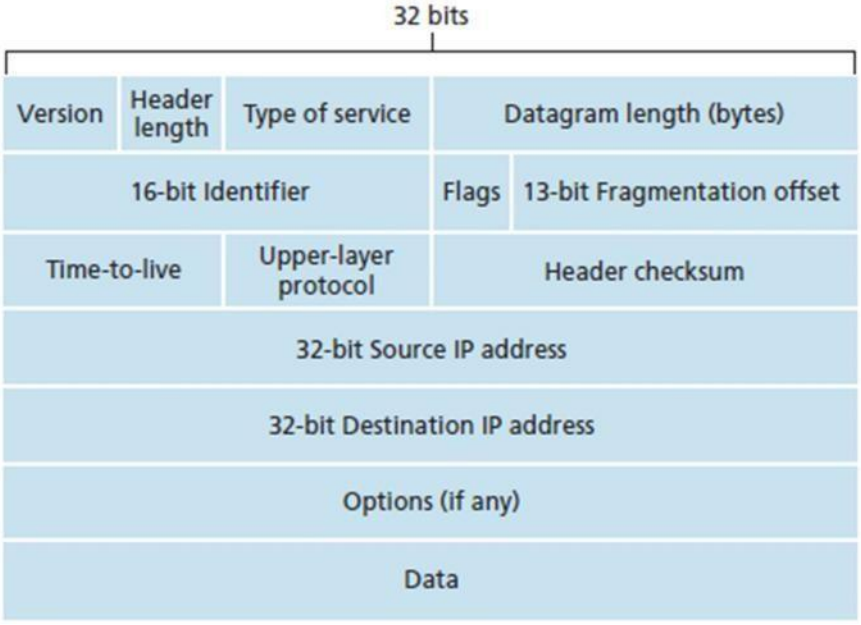


Fig. 7 IPv4 datagram format

- **Time-to-live:** Specifies how long the datagram is allowed to "live" on the network. Each router decrements the value of the TTL field (reduces it by one) prior to transmitting it. If the TTL field drops to zero, the datagram is assumed to have taken too long a route and isdiscarded.
- **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. For example, a value of 6 indicates that the data portion is passed to TCP, while a value of 17 indicates that the data is passed to UDP.
- **Header checksum:** The header checksum aids a router in detecting bit errors in a received IP datagram.
- **Source and destination IP addresses:** When a source creates a datagram, it inserts its IP address into the source IP address field and inserts the address of the ultimate destination into the destination IP address field.
- **Options:** The options fields allow an IP header to beextended.
- **Data (payload):** The data to be transmitted in the datagram, either an entire higher-layer message or a fragment of one.

IP addressing: introduction

- **IP address:** It is 32-bit identifier for host, routerinterface
- **Interface:** It is a connection between host/router and physicallink.
  - A router's typically have multiple interfaces
  - A host typically has one or two interfaces
- There is an IP addresses associated with each interface.
- **Subnets:** To determine the subnets, detach each interface from its host or router, creating islands of isolated networks, with interfaces terminating the end points of the isolated networks. Each of these isolated networks is called a subnet.
- **Subnet part**: high order bits defines subnet
- **Host part:** low order bits defines host.

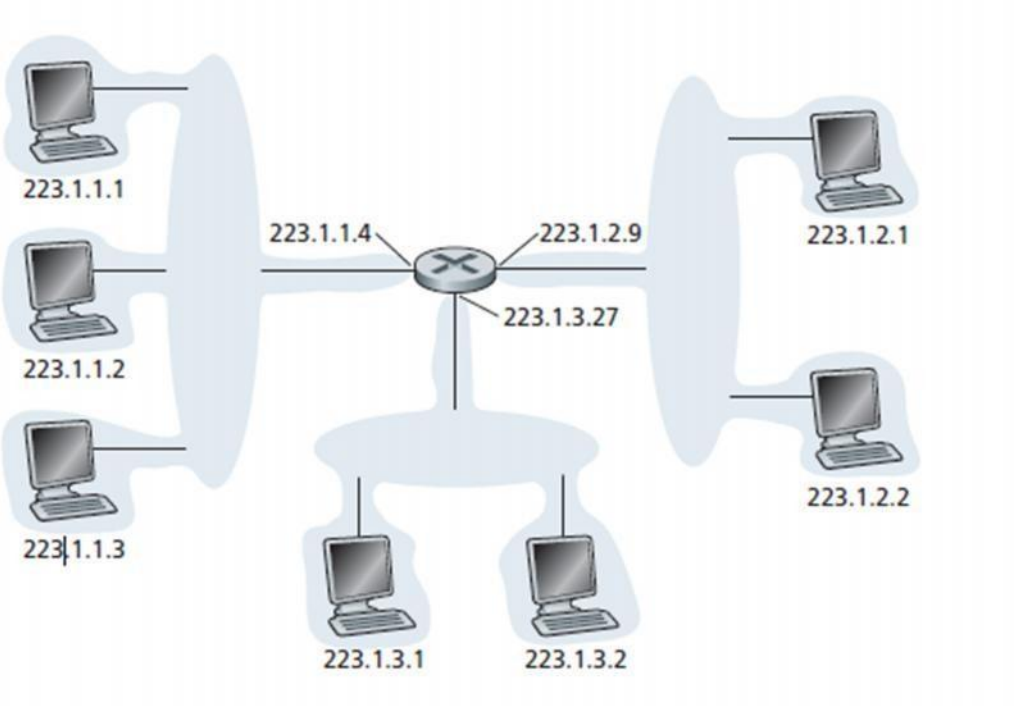


Fig. 8 Interface addresses and subnets

CIDR (Classless InterDomain Routing)

- Originally, IP addresses were assigned in four major address classes, A through D.
- Each of these classes allocates one portion of the 32-bit IP address format to identify a network gateway - the first 8 bits for class A, the first 16 for class B, and the first 24 for class C. The remainder identify hosts on that network.
- More than 16 million in class A, 65,535 in class B and 254 in class C. (Class D addresses identify multicast domains.)
- To illustrate the problems with the class system, consider that one of the most commonly used classes was Class B.
- An organization that needed more than 254 host machines (500 hosts) would often get a Class B license, even though it would have far fewer than 65,534 hosts.
- This resulted in most of the block of addresses allocated going unused.
- CIDR reduced the problem of wasted address space by providing a new and more flexible way to specify network addresses in routers.
- A single IP address can be used to designate many unique IP addresses with CIDR.
- A CIDR IP address looks like a normal IP address except that it ends with a slash followed by a number, called the IP network prefix. CIDR addresses reduce the size of routing tables and make more IP addresses available within organizations.

DHCP: Dynamic Host Configuration Protocol

- Dynamic Host Configuration Protocol is a protocol for assigning dynamic IP addresses to devices on a network.
- With dynamic addressing, a device can have a different IP address every time it connects to the network.
- In some systems, the device's IP address can even change while it is still connected. It allows reuse of addresses (only hold address while connected “on”). It also support mobile users who want to join network.

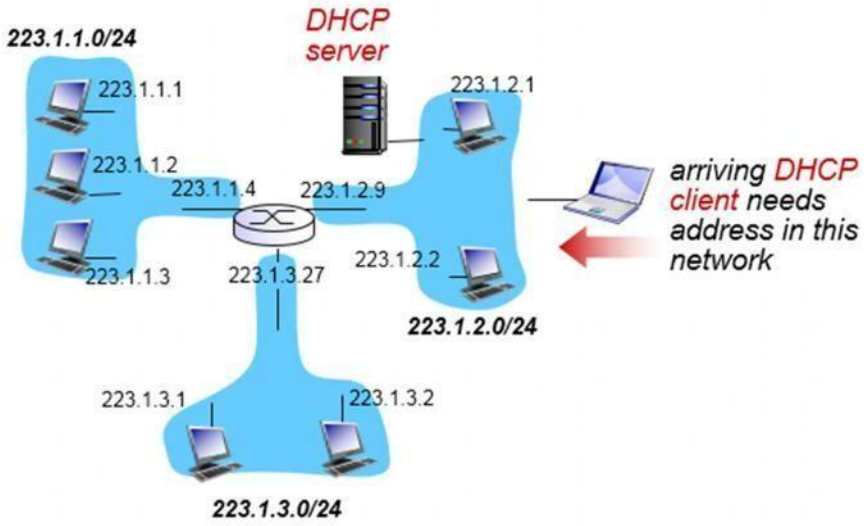


Fig. 9 DHCP client-server scenario

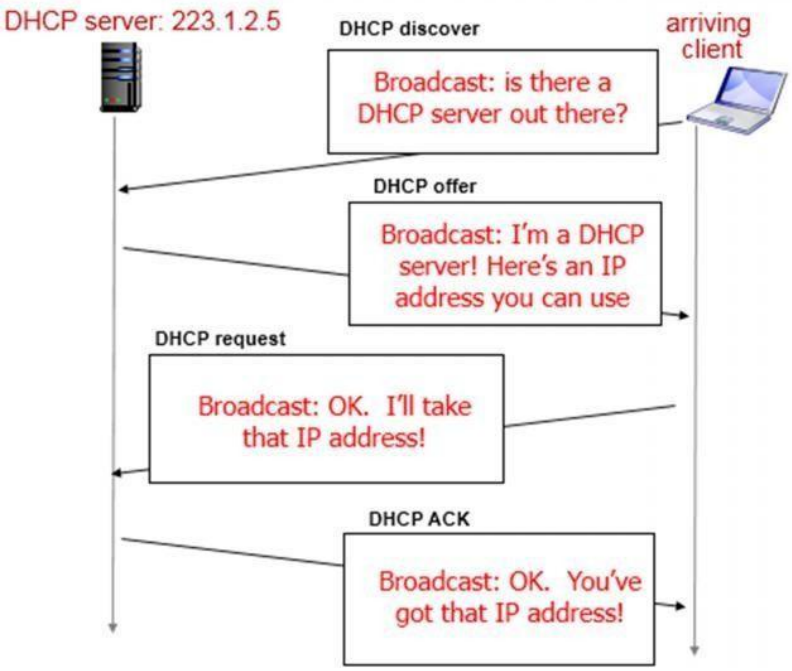


Fig. 10 DHCP client-server interaction

DHCP server discovery

- The first task of a newly arriving host is to find a DHCP server with which tointeract.
- This is done using a DHCP discover message, which a client sends within a UDP packet to port 67.
- The UDP packet is encapsulated in an IP datagram. But to whom should this datagram be sent? The host doesn't even know the IP address of the network to which it isattaching.
- Given this, 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 enviable 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 - the amount of time for which the IP address will be valid.

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.

Network Address Translation (NAT)

- The Internet has grown larger than anyone ever imagined it couldbe.
- Although the exact size is unknown, the current estimate is that there are about 100 million hosts and more than 350 million users actively on theInternet.
- In fact, the rate of growth has been such that the Internet is effectively doubling in size eachyear.
- So what does the size of the Internet have to do with NAT? For a computer to communicate with other computers and Web servers on the Internet, it must have an IPaddress.
- An IP address is a unique 32-bit number that identifies the location of your computer on a network.
- When IP addressing first came out, everyone thought that there were sufficiently of addresses to cover any need. Theoretically, you could have 4,294,967,296 unique addresses (232). The actual number of available addresses is smaller (somewhere between 3.2 and 3.3 billion) because of the way that the addresses are separated into classes, and because some addresses are set aside for multicasting, testing or other special uses.
- With the explosion of the Internet and the increase in home networks and business networks, the number of available IP addresses is simply not enough.
- The obvious solution is to redesign the address format to allow for more possible addresses. This is being developed (called IPv6), but will take several years to implement because it requires modification of the entire infrastructure of theInternet.

- This is where NAT (RFC 1631) comes to the rescue.
- Network Address Translation allows a single device, such as a router, to act as an agent between the Internet (or "public network") and a local (or "private") network.
- This means that only a single, unique IP address is required to represent an entire group of computers.

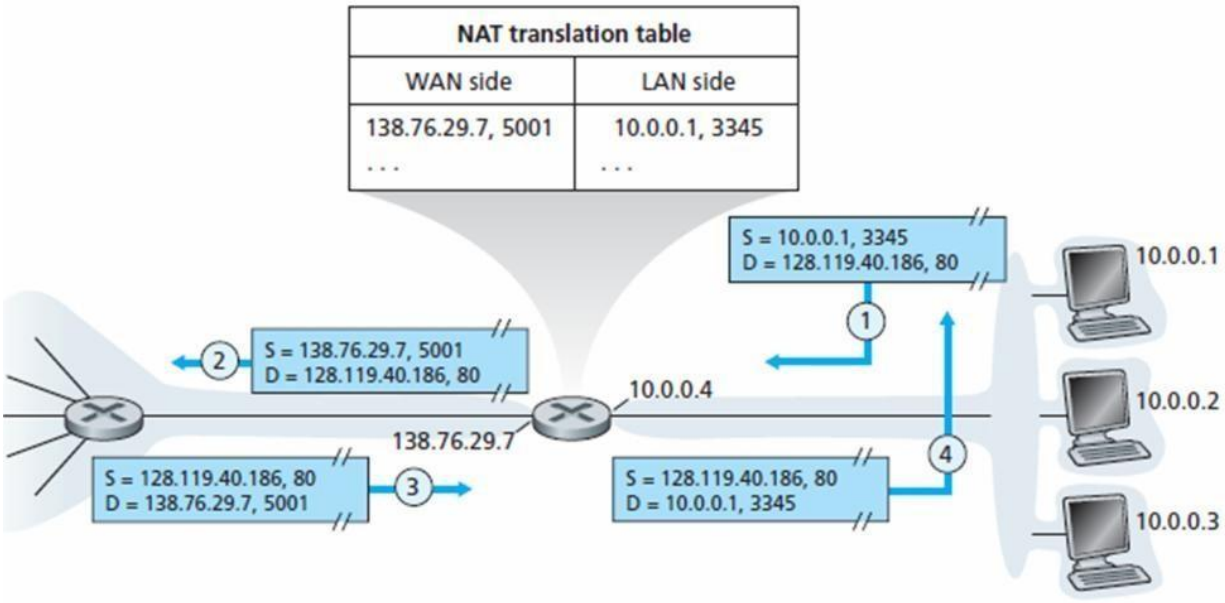


Fig. 11 Network address translation

ICMP: Internet Control Message Protocol

- When something unexpected occurs, the event is reported by the ICMP (Internet Control Message Protocol), which is also used to test the Internet.
- About a dozen types of ICMP messages are defined. The most important ones are listed below. Each ICMP message type is encapsulated in an IP packet.

Message Type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

- The DESTINATION UNREACHABLE message is used when the subnet or a router cannot locate the destination or when a packet with the DF bit cannot be delivered because a ''small-packet'' network stands in the way.
- The TIME EXCEEDED message is sent when a packet is dropped because its counter has reached zero.

- The PARAMETER PROBLEM message indicates that an illegal value has been detected in a header field.
- This problem indicates a bug in the sending host's IP software or possibly in the software of a router transited.
- The SOURCE QUENCH message was formerly used to throttle hosts that were sending too many packets. When a host received this message, it was expected to slowdown.
- The REDIRECT message is used when a router notices that a packet seems to be routed wrong. It is used by the router to tell the sending host about the probableerror.
- The ECHO and ECHO REPLY messages are used to see if a given destination is reachable andalive.
- Upon receiving the ECHO message, the destination is expected to send an ECHO REPLY message back.
- The TIMESTAMP REQUEST and TIMESTAMP REPLY messages are similar, except that the arrival time of the message and the departure time of the reply are recorded in thereply.

IPv6 Datagram Format

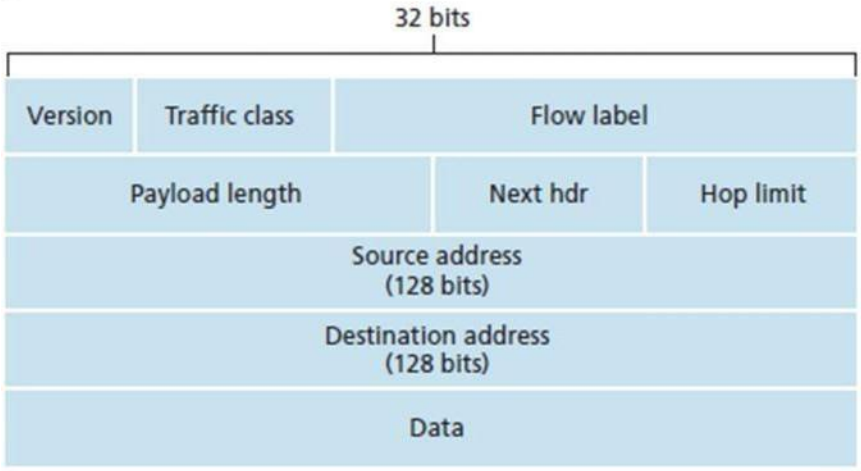


Fig. 12 IPv6 datagram format

- **Version:** The size of the Version field is 4 bits. The Version field shows the version of IP and is set to 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. The Traffic Class field indicates the IPv6 packet's class orpriority.
- **Flow Label**: The size of Flow Label field is 20 bits. The Flow Label field provide additional support for real-time datagram delivery and quality of service features. The purpose of Flow Label field is to indicate that this 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**: The size of the Payload Length field is 16 bits. The Payload Length field shows the length of the IPv6 payload,including the extension headers and the upper layer protocol data
- **Next Header**: The size of the Next Header field is 8 bits. The Next Header field shows either the type of the first extension (if any extension header is available) or the protocol in the upper layer such as TCP, UDP, or ICMPv6.
- **Hop Limit**: The size of the Hop Limit field is 8 bits The Hop Limit field shows the maximum number of routers the IPv6 packet can travel. This Hop Limit field is similar to IPv4 Timeto Live (TTL) field.

- **Source Address:** The size of the Source Address field is 128 bits. The Source Address field shows the IPv6 address of the source of the packet.
- **Destination Address:** The size of the Destination Address field is 128 bits. The Destination Address field shows the IPv6 address of the destination of the packet.
- **Data:** The data to be transmitted in the datagram, either an entire higher-layer message or a fragment of one.

Difference between IPv4 and IPv6

IPv4	IPv6
• IPv4 addresses are 32 bit length.	• IPv6 addresses are 128 bit length.
• Fragmentation is done by sender and forwarding routers.	• Fragmentation is done only by sender.
• No packet flow identification.	• Packet flow identification is available within the IPv6 header using the Flow Label field.
• Checksum field is available in header	• No checksum field in header.
• Options fields are available in header.	• No option fields, but Extension headers are available.
• Address Resolution Protocol (ARP)is available to map IPv4 addresses to MAC addresses.	• Address Resolution Protocol (ARP) is replaced with Neighbour Discovery Protocol.
• Broadcast messages are available.	• Broadcast messages are not available.
• Manual configuration (Static) of IP addresses or DHCP (Dynamic configuration) is required to configure IP addresses.	• Auto-configuration of addresses is available.

The Link-State (LS) Routing Algorithm (Dijkstra's algorithm)

- Dijkstra's algorithm computes the least-cost path from one node (the source, which we will refer to as u) to all other nodes in the network.
- Dijkstra's algorithm is iterative and has the property that after the k^{th} iteration of the algorithm, the least-cost paths are known to k destination nodes, and among the least-cost paths to all destination nodes, these k paths will have the k smallest costs.
- Let us define the following notation:
  - D(v): cost of the least-cost path from the source node to destination v as of this iteration of the algorithm.
  - p(v): previous node (neighbor of v) along the current least-cost path from the source to v.
  - N' : subset of nodes; v is in N' if the least-cost path from the source to v is definitively known.
- The global routing algorithm consists of an initialization step followed by a loop.
- The number of times the loop is executed is equal to the number of nodes in the network.
- Upon termination, the algorithm will have calculated the shortest paths from the source node u to every other node in the network.
- As an example, let's consider the network in Figure 13 and compute the least-cost paths from u to all possible destinations.

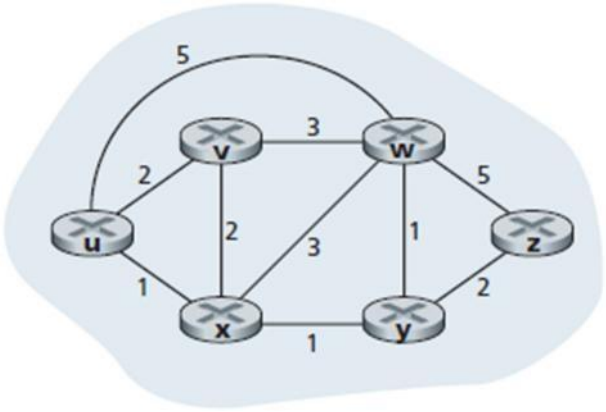


Fig. 13 Abstract graph model of a computer network

```
1  Initialization:
2    N' = {u}
3    for all nodes v
4      if v is a neighbor of u
5        then D(v) = c(u,v)
6      else D(v) = ∞
7
8  Loop
9    find w not in N' such that D(w) is a minimum
10   add w to N'
11   update D(v) for each neighbor v of w and not in N':
12         D(v) = min( D(v), D(w) + c(w,v) )
13   /* new cost to v is either old cost to v or known
14    least path cost to w plus cost from w to v */
15 until N'= N
```

- Let's consider the few first steps in detail.
- In the initialization step, the currently known least-cost paths from u to its directly attached neighbours, v, x, and w, are initialized to 2, 1, and 5, respectively. Note in particular that the cost to w is set to 5 (even though we will soon see that a lesser-cost path does indeed exist) since this is the cost of the direct (one hop) link from u to w. The costs to y and z are set to infinity because they are not directly connected to u.
- In the first iteration, we look among those nodes not yet added to the set N' and find that node with the least cost as of the end of the previous iteration. That node is x, with a cost of 1, and thus x is added to the set N'. Line 12 of the LS algorithm is then performed to update D(v) for all nodes v, yielding the results shown in the second line (Step 1) in below table. The cost of the path to v is unchanged. The cost of the path to w (which was 5 at the end of the initialization) through node x is found to have a cost of 4. Hence this lower-cost path is selected and w's predecessor along the shortest path from u is set to x. Similarly, the cost to y (through x) is computed to be 2, and the table is updated accordingly.
- In the second iteration, nodes v and y are found to have the least-cost paths (2), and we break the tie arbitrarily and add y to the set N' so that N' now contains u, x, and y. The cost to the remaining

nodes not yet in N', that is, nodes v, w, and z are updated via line 12 of the LS algorithm, yielding the results shown in the third row in the below table.

- And so on. . . .
- When the LS algorithm terminates, we have, for each node, its predecessor along the least-cost path from the source node.
- For each predecessor, we also have its predecessor, and so in this manner we can construct the entire path from the source to all destinations.
- The forwarding table in a node, say node u, can then be constructed from this information by storing, for each destination, the next-hop node on the least-cost path from u to thedestination.
- Figure 14. Shows the resulting least-cost paths for u for the network in Figure 13.

step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

Table: Running the link-state algorithm on the network in Figure 13

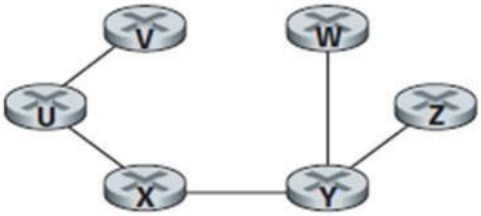


Fig. 14 Least cost path for nodule u

The Distance-Vector (DV) Routing Algorithm

- Distance-vector (DV) algorithm is iterative, asynchronous, and distributed.
- It is distributed in that each node receives some information from one or more of its directly attached neighbours, performs a calculation, and then distributes the results of its calculation back to its neighbours.
- It is iterative in that this process continues on until no more information is exchanged between neighbours.
- The algorithm is asynchronous in that it does not require all of the nodes to operate in lockstep with each other.
- Let dx(y) be the cost of the least-cost path from node x to node y. Then the least costs are related by the celebrated Bellman-Ford equation:

$$d_x(y) = \min_v\{c(x,v) + d_v(y)\}$$

- where the $\min_v$ in the equation is taken over all of x's neighbours. Indeed, after traveling from x to v, if we then take the least-cost path from v to y, the path cost will be c(x,v) + dv(y).

- Since we must begin by traveling to some neighbour v, the least cost from x to y is the minimum of c(x,v) + dv(y) taken over all neighbours v.

```
1  Initialization:
2     for all destinations y in N:
3        Dx(y) = c(x,y)    /* if y is not a neighbor then c(x,y) = ∞ */
4     for each neighbor w
5        Dw(y) = ? for all destinations y in N
6     for each neighbor w
7        send distance vector Dx = [Dx(y): y in N] to w
8
9  loop
10    wait (until I see a link cost change to some neighbor w or
11          until I receive a distance vector from some neighbor w)
12
13    for each y in N:
14       Dx(y) = minv{c(x,v) + Dv(y)}
15
16    if Dx(y) changed for any destination y
17       send distance vector Dx = [Dx(y): y in N] to all neighbors
18
19 forever
```

- Figure 15 illustrates the operation of the DV algorithm for the simple three node network shown at the top of the figure.
- The operation of the algorithm is illustrated in a synchronous manner, where all nodes simultaneously receive distance vectors from their neighbours, compute their new distance vectors, and inform their neighbours if their distance vectors have changed.
- The leftmost column of the figure displays three initial routing tables for each of the threenodes.
- For example, the table in the upper-left corner is node x's initial routingtable.
- Within a specific routing table, each row is a distance vector - specifically, each node's routing table includes its own distance vector and that of each of itsneighbours.
- Thus, the first row in node x's initial routing table is $D_x = [D_x(x), D_x(y), D_x(z)] = [0, 2, 7]$.
- The second and third rows in this table are the most recently received distance vectors from nodes y and z, respectively.
- Because at initialization node x has not received anything from node y or z, the entries in the second and third rows are initialized to infinity.
- After initialization, each node sends its distance vector to each of its twoneighbours.
- This is illustrated in Figure 15 by the arrows from the first column of tables to the second column of tables.
- For example, node x sends its distance vector $D_x = [0, 2, 7]$ to both nodes y and z. After receiving the updates, each node recomputes its own distance vector.
- For example, node x computes

$$D_x(x) = 0$$
$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} = \min\{2 + 0, 7 + 1\} = 2$$
$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} = \min\{2 + 1, 7 + 0\} = 3$$

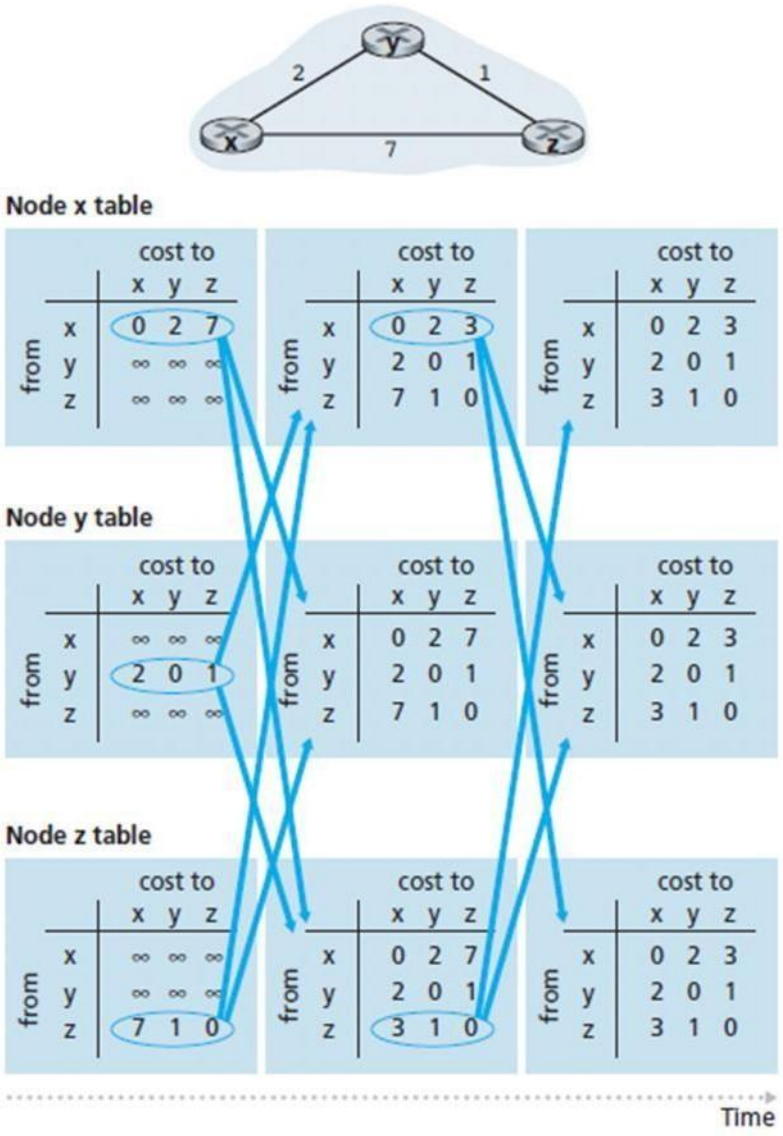


Fig. 15 Distance-vector (DV) algorithm

- The second column therefore displays, for each node, the node's new distance vector along with distance vectors just received from its neighbours.
- Note, that node x's estimate for the least cost to node z, Dx(z), has changed from 7 to 3.
- Also note that for node x, neighbouring node y achieves the minimum in line 14 of the DV algorithm; thus at this stage of the algorithm, we have at node x that v*(y) =y and v*(z) =y.
- After the nodes recomputes their distance vectors, they again send their updated distance vectors to their neighbours (if there has been a change).
- This is illustrated in Figure 15 by the arrows from the second column of tables to the third column of tables.
- Note that only nodes x and z send updates: node y's distance vector didn't change so node y doesn't send an update.

- After receiving the updates, the nodes then recomputes their distance vectors and update their routing tables, which are shown in the third column.
- The process of receiving updated distance vectors from neighbours, recomputing routing table entries, and informing neighbours of changed costs of the least-cost path to a destination continues until no update messages are sent.
- At this point, since no update messages are sent, no further routing table calculations will occur and the algorithm will enter a quiescent state; that is, all nodes will be performing the wait in Lines 10–11 of the DV algorithm.
- The algorithm remains in the quiescent state until a link costchanges.

Comparison of (Difference between) LS and DV Routing Algorithms

Distance Vector Protocol	Link state protocol
Entire routing table is sent as an update	Updates are incremental & entire routing table is not sent as update
Distance vector protocol send periodic update at every 30 or 90 second	Updates are triggered not periodic
Update are broadcasted	Updates are multicasted
Updates are sent to directly connected neighbour only	Update are sent to entire network & to just directly connected neighbour
Routers don't have end to end visibility of entire network.	Routers have visibility of entire network of that area only.
It is prone to routing loops	No routing loops

The Count to Infinity problem

- Distance vector routing works in theory but has a serious drawback inpractice.
- Consider a router whose best route to destination X is large.
- If on the next exchange neighbour A suddenly reports a short delay to X, the router just switches over to using line to A to send traffic to X.
- Suppose A is down initially and all the other routers know this. In other words, they have all recorded the delay to A as infinity.
- When A comes up, the other routers learn about it via the vectorexchanges.
- At the time of the first exchange, B learns that its left neighbour has zero delay to A.
- B now makes an entry in its routing table that A is one hop away to theleft.
- All the other routers still think that A is down. At this point, the routing table entries for A are as shown in the second row of Figure 16 (a).
- On the next exchange, C learns that B has a path of length 1 to A, so it updates its routing table to indicate a path of length 2, but D and E do not hear the good news untillater.
- Clearly, the good news is spreading at the rate of one hop per exchange.

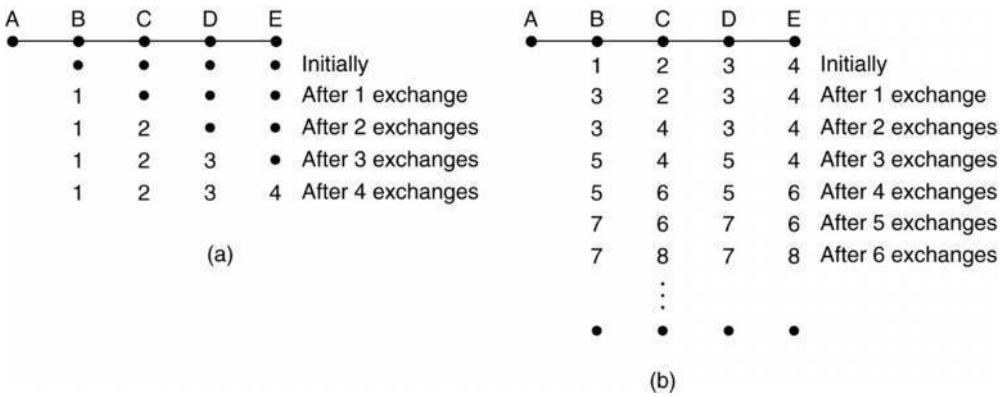


Fig. 16: The Count to infinity problem

- Now let us consider the situation of Figure 16 (b), in which all the lines and routers are initially up. Routers B, C, D, and E have distances to A of 1, 2, 3, and 4, respectively.
- Suddenly A goes down, or alternatively, the line between A and B is cut, which is effectively the same thing from B's point of view.
- At the first packet exchange, B does not hear anything from A.
- Fortunately, C says: Do not worry; I have a path to A of length 2.
- Little does B know that C's path runs through B itself. For all B knows, C might have ten lines all with separate paths to A of length 2.
- As a result, B thinks it can reach A via C, with a path length of 3. D and E do not update their entries for A on the first exchange.
- On the second exchange, C notices that each of its neighbours claims to have a path to A of length 3.
- It picks one of them at random and makes its new distance to A 4, as shown in third row of Figure 16(b).
- Subsequent exchanges produce the history shown in the rest of Figure 16(b).
- From this figure, it should be clear why bad news travels slowly: no router ever has a value more than one higher than the minimum of all its neighbours.
- Gradually, all routers work their way up to infinity, but the number of exchanges required depends on the numerical value used for infinity.
- For this reason, it is wise to set infinity to the longest path plus 1.
- Not entirely surprisingly, this problem is known as the count-to-infinity problem.

Hierarchical Routing

- As networks grow in size, the router routing tables grow proportionally.
- Not only is router memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them.
- At a certain point the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network.
- When hierarchical routing is used, the routers are divided into what called **regions**, with each router knowing all the details about how to route packets to destinations within its own region, but knowing nothing about the internal structure of other regions.

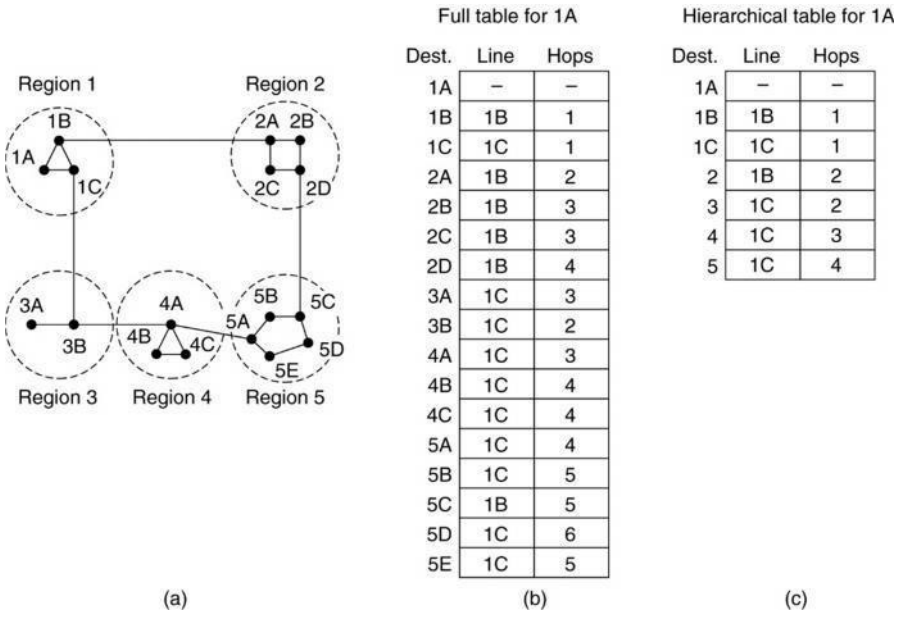


Fig. 2: Hierarchical Routing

- Figure 17 gives a quantitative example of routing in a two-level hierarchy with five regions. The full routing table for router 1A has 17 entries, as shown in Figure 17(b).
- When routing is done hierarchically, as in Figure 17 (c), there are entries for all the local routers as before, but all other regions have been condensed into a single router, so all traffic for region 2 goes via the 1B -2A line, but the rest of the remote traffic goes via the 1C -3B line. Hierarchical routinghas reduced the table from 17 to 7 entries.
- As the ratio of the number of regions to the number of routers per region grows, the savings in table space increase.
- Unfortunately, these gains in space are not free. There is a penalty to be paid, and this penalty is in the form of increased path length.
- For example, the best route from 1A to 5C is via region 2, but with hierarchical routing all traffic to region 5 goes via region 3, because that is better for most destinations in region5.
- When a single network becomes very large, an interesting question is: How many levels should the hierarchy have? For example, consider a subnet with 720 routers.
- If there is no hierarchy, each router needs 720 routing table entries. If the subnet is partitioned into 24 regions of 30 routers each, each router needs 30 local entries plus 23 remote entries for a total of 53 entries.

Broadcast Routing

- In some applications, hosts need to send messages to many or all otherhosts.
- For example, a service distributing weather reports, stock market updates, or live radio programs might work best by broadcasting to all machines and letting those that are interested read the data.
- Sending a packet to all destinations simultaneously is called broadcasting.
- **First** broadcasting method that requires no special features from the subnet is for the source to simply send a distinct packet to each destination.

- Not only is the method wasteful of bandwidth, but it also requires the source to have a complete list of all destinations. In practice this may be the only possibility, but it is the least desirable of the methods.

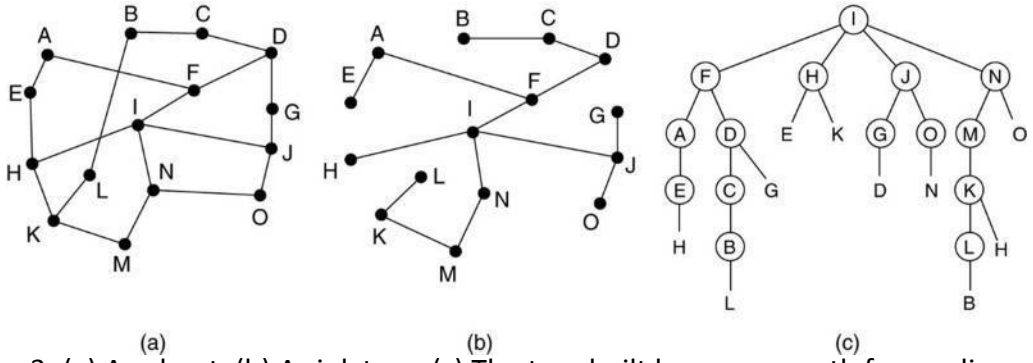


Fig. 3: (a) A subnet. (b) A sink tree. (c) The tree built by reverse path forwarding

- Flooding is another Second method. Although flooding is ill-suited for ordinary point-to-point communication, for broadcasting it might rate serious consideration, especially if none of the methods described below are applicable.
- The problem with flooding as a broadcast technique is the same problem it has as a point-to-point routing algorithm: it generates too many packets and consumes too muchbandwidth.
- A third algorithm is multi destination routing.
- If this method is used, each packet contains either a list of destinations or a bit map indicating the desired destinations.
- When a packet arrives at a router, the router checks all the destinations to determine the set of output lines that will be needed.
- The router generates a new copy of the packet for each output line to be used and includes in each packet only those destinations that are to use the line.
- A fourth broadcast algorithm makes explicit use of the sink tree for the router initiating the broadcast-or any other convenient spanning tree for that matter.
- A spanning tree is a subset of the subnet that includes all the routers but contains no loops.
- If each router knows which of its lines belong to the spanning tree, it can copy an incoming broadcast packet onto all the spanning tree lines except the one it arrived on.
- Fifth broadcast algorithm is reverse path forwarding, is remarkably simple once it has been pointed out.
- When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the line that is normally used for sending packets to the source of thebroadcast.
- If so, there is an excellent chance that the broadcast packet itself followed the best route from the router and is therefore the first copy to arrive at the router.
- This being the case, the router forwards copies of it onto all lines except the one it arrived on.

Multicast Routing

- Sending a message to a group is called multicasting, and its routing algorithm is called multicast routing.
- Multicasting requires group management. Some way is needed to create and destroy groups, and to allow processes to join and leave groups.

- To do multicast routing, each router computes a spanning tree covering all otherrouters.
- For example, in Figure (a) we have two groups, 1 and 2.
- Some routers are attached to hosts that belong to one or both of these groups, as indicated in the figure.
- A spanning tree for the leftmost router is shown in Figure (b).
- When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, removing all lines that do not lead to hosts that are members of thegroup.
- In our example, Figure (c) shows the pruned spanning tree for group 1.

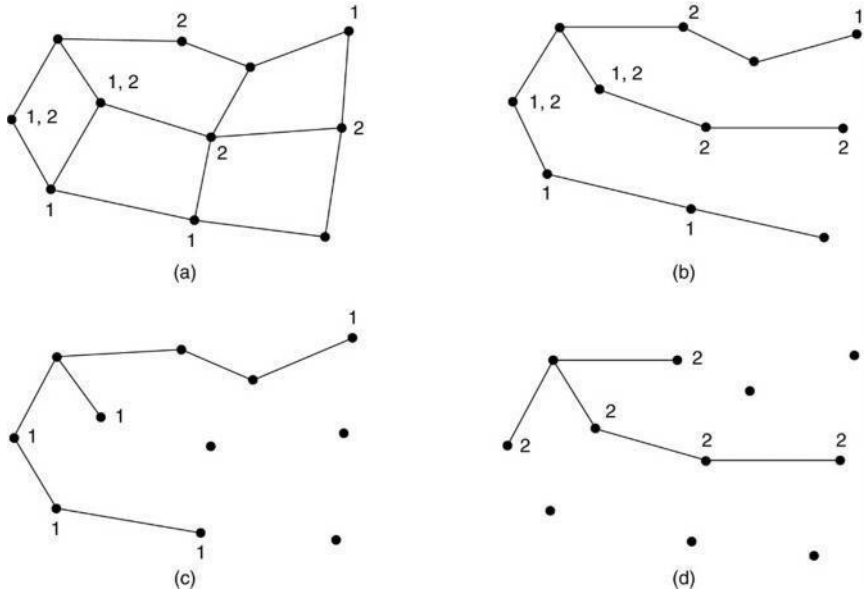


Fig. 49: (a) A network. (b) A spanning tree for the leftmost router

- Similarly, Figure (d) shows the pruned spanning tree for group 2. Multicast packets are forwarded only along the appropriate spanning tree.

Intra-AS Routing

- It is also known as interior gateway protocols (IGP)
- Most common intra-AS routing protocols:
    1. RIP: Routing Information Protocol
    2. OSPF: Open Shortest Path First
    3. IGRP: Interior Gateway Routing Protocol

RIP (Routing Information Protocol)

- The Routing Information Protocol (RIP) defines a way for routers, which connect networks using the Internet Protocol (IP), to share information about how to route traffic amongnetworks.
- Each RIP router maintains a routing table, which is a list of all the destinations (networks) it knows how to reach, along with the distance to that destination.
- RIP uses a distance vector algorithm to decide which path to put a packet on to get to its destination.
- It stores in its routing table the distance for each network it knows how to reach, along with the address of the "next hop" router - another router that is on one of the same networks - through which a packet has to travel to get to that destination.

- If it receives an update on a route and the new path is shorter, it will update its table entry with the length and next-hop address of the shorter path; if the new path is longer, it will wait through a "hold-down" period to see if later updates reflect the higher value as well, and only update the table entry if the new, longer path is stable.
- Using RIP, each router sends its entire routing table to its closest neighbours every 30 seconds. (The neighbours are the other routers to which this router is connected directly - that is, the other routers on the same network segments this router ison.)
- The neighbours in turn will pass the information on to their nearest neighbours, and so on, until all RIP hosts within the network have the same knowledge of routing paths, a state known as convergence.

OSPF (Open Shortest Path First)

- The Internet is made up of a large number of **Autonomous Systems** (**AS**).
- A routing algorithm within an AS is called an interior gateway protocol; an algorithm for routing between AS is called an exterior gateway protocol.
- Many of the ASes in the Internet are themselves large and nontrivial tomanage.
- OSPF allows them to be divided into numbered areas, where an area is a network or a set of contiguous networks.
- Areas do not overlap but need not be exhaustive, that is, some routers may belong to no area. An area is a generalization of a subnet.
- Every AS has a backbone area, called area 0.
- All areas are connected to the backbone, possibly by tunnels, so it is possible to go from any area in the AS to any other area in the AS via the backbone.
- Each router that is connected to two or more areas is part of the backbone. As with other areas, the topology of the backbone is not visible outside thebackbone.

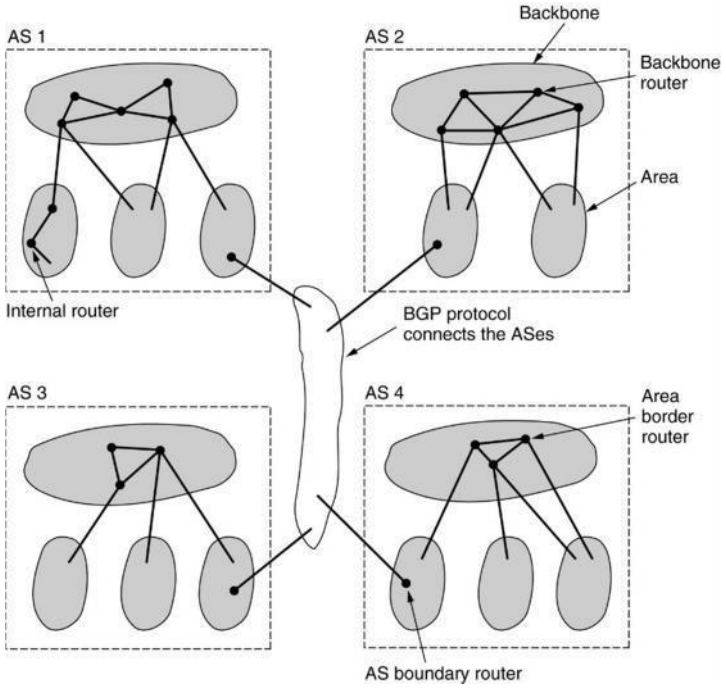


Fig. 5: The relation between ASes, backbones, and areas in OSPF

- Within an area, each router has the same link state database and runs the same shortest path algorithm.
- Its main job is to calculate the shortest path from itself to every other router in the area, including the router that is connected to the backbone, ofwhich there must be at least one.
- A router that connects to two areas needs the databases for both areas and must run the shortest path algorithm for each one separately.
- This algorithm forces a star configuration on OSPF with the backbone being the hub and the other areas being spokes. Packets are routed from source to destination ''as is.''
- They are not encapsulated or tunneled, unless going to an area whose only connection to the backbone is a tunnel. Figure shows part of the Internet with ASes and areas.

OSPF distinguishes four classes of routers:

1. Internal routers are wholly within one area.
2. Area border routers connect two or more areas.
3. Backbone routers are on the backbone.
4. AS boundary routers talk to routers in other ASes.

Area border router (ABR)

- An area border router (ABR) is a router that connects one or more areas to the main backbone network.
- It is considered a member of all areas it is connected to.
- An ABR keeps multiple copies of the link-state database in memory, one for each area to which that router is connected.

Autonomous system boundary router (ASBR)

- An autonomous system boundary router (ASBR) is a router that is connected to more than one Routing protocol and that exchanges routing information with routers in otherprotocols.
- ASBRs typically also run an exterior routing protocol (e.g., BGP), or use static routes, or both.
- An ASBR is used to distribute routes received from other, external ASs throughout its own autonomous system.

Internal router (IR)

- An internal router is a router that has OSPF neighbour relationships with interfaces in the same area. An internal router has all its interfaces in a single area.

Backbone router (BR)

- The backbone routers accept information from the area border routers in order to compute the best route from each backbone router to every other router.
- This information is propagated back to the area border routers, which advertise it within their areas.

Comparison between RIP OSPF and BGP

RIP	OSPF	BGP
RIP is intra domain routing protocol used with in the autonomous system	OSPF is also intra domain routing protocol used with in the autonomous system	It is inter domain routing protocol used between the autonomous system
RIP is used for Small networks with maximum number of hops 16	OSPF is used in large autonomous system with no limitation	The BGP protocol is used for very large-scale networks
RIP uses Distance Vector	OSPF uses Link State	BGP uses Path Vector

RIP send entire routing update to all directly connected interface	OSPF send multicast Hello packet to the neighbours, to create session	BGP send Open packet to the neighbours to create session
RIP use Bellman ford Algorithm	OSPF use Dijikstra Algorithm	BGP use Path-Vector Routing

Consider a router that interconnects three subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix 23.1.17/24. Also suppose that Subnet 1 is required to support at least 60 interfaces, Subnet 2 is to support at least 90 interfaces, and Subnet 3 is to support at least 12 interfaces. Provide three network addresses (of the form a.b.c.d/x) that satisfy these constraints.

- For Subnet1 we have to support at least 60 interfaces and 2^6 >= 60 so the prefix for subnet1 is 32-6 = 26 for **subnet1 = 23.1.17.x/26**
- For Subnet2 we have to support at least 90 interfaces and 2^7 >= 90 so the prefix for subnet2 is 32-7 = 25 , and so **subnet2 = 23.1.17.y/25**
- For Subnet3 we have to support at least 12 interfaces and 2^4 >= 12 so the prefix for subnet3 is 32-4 = 28 , and so **subnet3 = 23.1.17.z/28**
- Now find the values for x,y and z.
  - subnet 1 **23.1.17.0/26**
  - subnet 2 **23.1.17.128/25**
  - subnet 3 **23.1.17.64/28**

Suppose datagrams are limited to 1,500 bytes (including header) between source Host A and destination Host B. Assuming a 20-byte IP header, how many datagrams would be required to send an MP3 consisting of 5 million bytes? Explain how you computed your answer.

Given: IP Header size = 20 bytes
Datagram Size = 1500 bytes
We know: TCP Header size = 20 bytes
So to find the data contain in each datagram we need to deduct IP and TCP Header that is
1500 - 20 - 20 = 1460 bytes
Each datagram can carry maximum 1460 bytes.
So we number of datagrams required to send 5 million bytes =5000000 / 1460 = 3424.66
So we need **3425** datagrams to carry 5 million bytes.