CO7219:
Internet and Cloud Computing

3. Internetworking

3.1 Internetworking

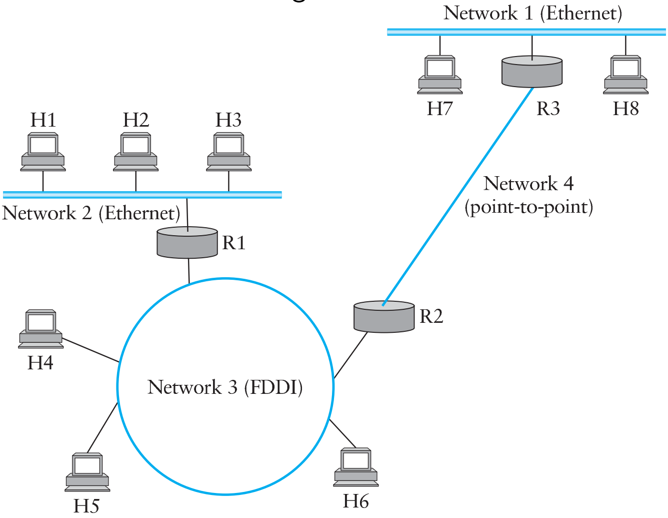
- A single network can be built using point-to-point links, shared media (Ethernet, WLAN), and switches.
- We want to interconnect different networks and build an **internetwork**.
- Two important problems must be addressed:
  - **Heterogeneity**: The individual networks may be based on very different technologies.
  - **Scale**: The Internet has roughly doubled its size each year for 20 years. How can one do **routing** and **addressing** efficiently for millions of nodes?

3.1.1 What is an Internetwork?

- An **internetwork** or **internet** is an arbitrary collection of networks (also called **physical networks**) interconnected to provide some sort of host-to-host packet delivery service.
  - An internet is a "network of networks".
- The **Internet** (with capital "I") is the widely used, global internetwork to which most networks are connected.
- The nodes that interconnect the networks are called **routers** (or sometimes **gateways**).

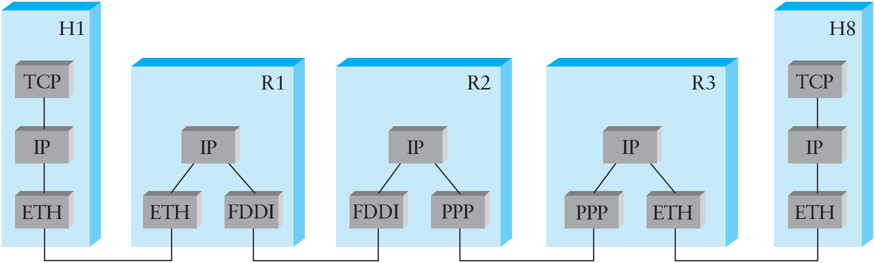
Example

- An internetwork consisting of 4 networks:



Internet Protocol (IP)

- The Internet Protocol is the key tool used today to build scalable, heterogeneous internetworks.
- IP runs on all nodes (hosts and routers) in a collection of networks and defines the infrastructure that allows them to function as a single logical network.



3.1.2 IP Service Model

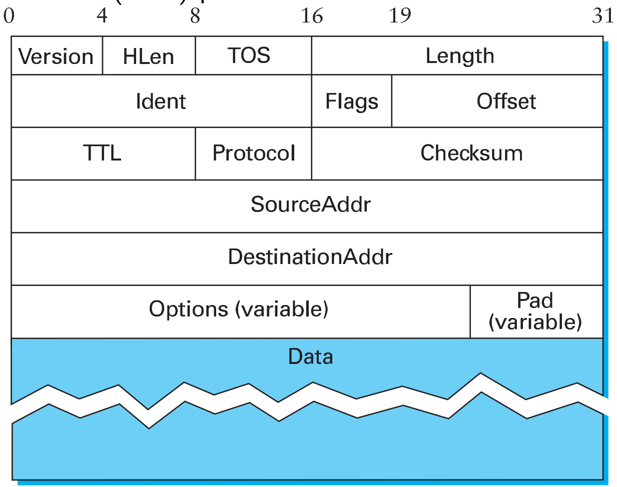
- **Datagram Delivery**
  - A datagram is a type of packet that is sent in a connectionless manner over a network. It carries enough information to let the network forward it to its correct destination.
  - “Connectionless” means that there is no advance setup mechanism to tell the network what to do when the packet arrives.
  - The network makes its **best effort** to get the packet to its destination. If something goes wrong (packet gets lost, corrupted, misdelivered), the network does nothing (**unreliable** service).
  - Packets can get delivered out of order, or more than once.

Connectionless Best-Effort Service

- **Advantages:**
  - Simplest service one could ask from a network.
  - Asking for a reliable packet delivery service would mean that a lot of extra functionality has to be put into the routers.
  - Best-effort service allows to keep the routers as simple as possible (one of the design goals of IP).
  - Enables IP to "run over anything". Today IP runs over many network technologies that did not even exist when IP was invented.
- Higher-level protocols and applications that run over IP need to be aware of all the possible failure modes.

IP Packet Format

- IP version 4 (IPv4) packet format:



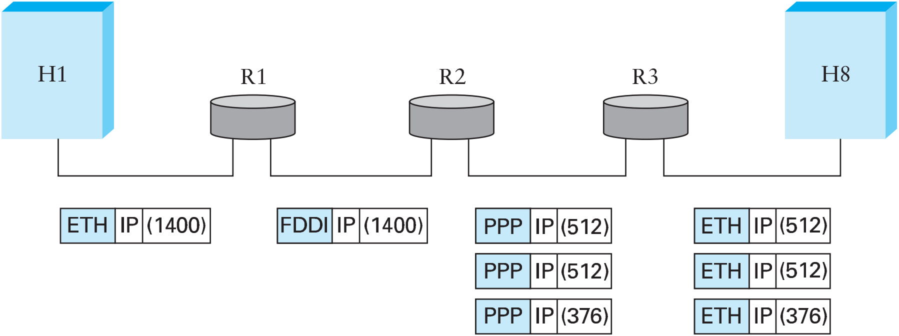
IP Header Fields

- Version: IP version number (4 in IPv4)
- HLen: Length of Header (in 32-bit words), normally 5
- TOS: Type of Service
- Length: Length in bytes (data + header), max. 65536
- Ident, Flags, Offset: used for fragmentation and reassembly
- TTL: Time to Live (number of hops before packet is discarded, default 64)
- Protocol: Demultiplexing key (e.g. TCP = 6, UDP = 17)
- Checksum: Internet checksum of header
- SourceAddr, Destination Addr: IP addresses of src/dest
- Options: rarely used (present only if HLen > 5)

Fragmentation and Reassembly

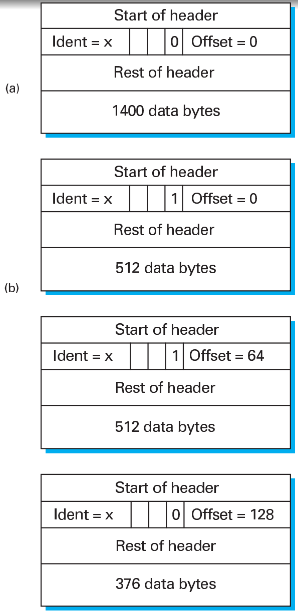
- **Problem:** IP allows packets of size up to 64 KB, but each network technology has its own limits on the size of the data in a packet: 1500 bytes for Ethernet, 4500 bytes for FDDI, etc.
- When an IP packet is to be sent over a network whose MTU (maximum transmission unit) is smaller than the size of that packet, the packet gets **fragmented** (split into smaller packets).
- The packet fragments are put together (**reassembly**) at the receiving host. All fragments originating from the same packet have the same identifier in the Ident field.
- The Ident field is chosen by the sending host and is intended to be unique among all datagrams from this source to the destination over a reasonable time period.

Fragmentation Example



- Remarks:
  - Each fragment is itself a self-contained IP datagram, transmitted independent of the other fragments.
  - Each IP datagram is reencapsulated (in a link-layer frame) for each physical network over which it travels.

Headers of Fragments



(a)

Start of header

Ident = x | | | 0 | Offset = 0

Rest of header

1400 data bytes

(b)

Start of header

Ident = x | | | 1 | Offset = 0

Rest of header

512 data bytes

Start of header

Ident = x | | | 1 | Offset = 64

Rest of header

512 data bytes

Start of header

Ident = x | | | 0 | Offset = 128

Rest of header

376 data bytes

(a) unfragmented packet

(b) 3 resulting fragments

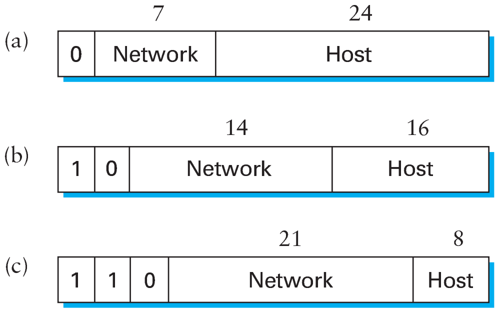
Remarks:

- In all fragments except the last, the M bit of the Flags field is set to 1.
- The Offset field gives the offset in the original packet where the data of this fragment begins (in multiples of 8 bytes).
- Note that a fragment can be split into smaller fragments if necessary.

3.1.3 Global Addresses

- In the IP service model, we need a global addressing scheme in which no two hosts have the same address (**global uniqueness**).
- Ethernet addresses are globally unique, but they are **flat** and provide no clues to routing protocols.
- IP addresses are **hierarchical**: They are made up of several parts that correspond to some sort of hierarchy.
  - Think of part of the IP address as identifying a network in the internetwork, and the rest of the IP address as identifying a host inside that network.
- Routers are attached to two or more networks: They have an address on each of these networks. Think of IP addresses as belonging to network interfaces rather than to hosts.

“Classful” IP Addresses



(a) 7 24

0 Network Host

(b) 14 16

1 0 Network Host

(c) 21 8

1 1 0 Network Host

(a) class A WANs

(b) class B site/campus

(c) class C LANs

- IP addresses are 32 bits long, written in the form 143.210.72.129 (i.e. each byte is given in decimal).

3.1.4 Datagram Forwarding in IP

- **Forwarding** is the process where an IP router takes a packet from an input and sends it out on the appropriate output.
- Every IP datagram contains the IP address of the destination, the "network part" of the address identifying a physical network that is part of the Internet.
- All hosts and routers that share the same network part of their address are connected to the same physical network, and can thus communicate with each other by sending frames over that network.
- Each physical network that is part of the Internet contains at least one router that is connected to at least one other physical network.

Forwarding an IP Datagram

- Any node – host or router – first compares the network part of the destination address with the network parts of the addresses of all its interfaces.
- If a match is found, the destination lies on the same physical network as the interface, and the packet can be delivered to the destination over that network.
- Otherwise, the node will send the packet to some (other) router, the **next hop** router:
  - The router finds the next hop by consulting its forwarding table (conceptually, this table contains a list of ⟨NetworkNum,NextHop⟩ pairs).
  - Normally, there is a **default router** that is used if no entry in the forwarding table matches the destination network. Hosts may have only a default router.

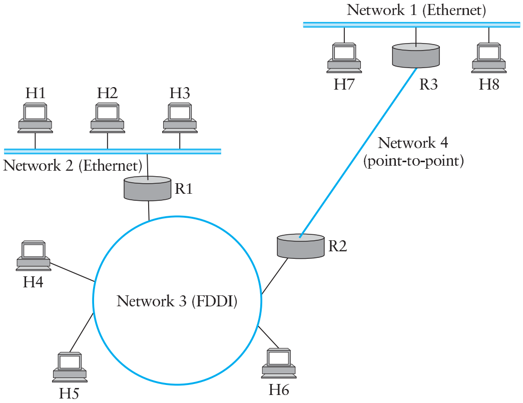
Forwarding on a Router

```
if (destination NetworkNum = NetworkNum of one of my interfaces)
then
    deliver packet to destination over that interface
else
    if (destination NetworkNum is in my forwarding table)
    then
        deliver packet to corresponding NextHop router
    else
        deliver packet to default router
    fi
fi
```

Forwarding on a Host

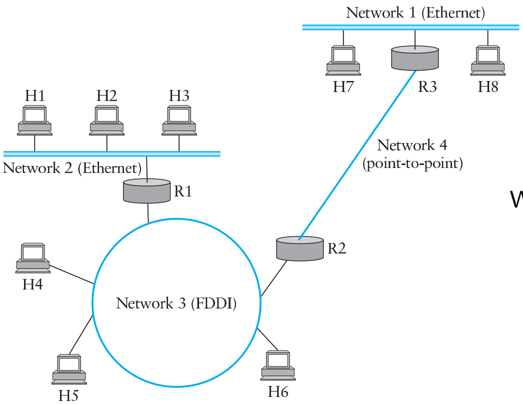
```
if (destination NetworkNum = my NetworkNum)
then
    deliver packet to destination directly
else
    deliver packet to default router
fi
```

Example 1



- If H1 sends to H2, it can send the packet directly since H1 and H2 are on the same physical network (so H2 has the same network part in the IP address).

Example 2



R2's forwarding table:

NetworkNum	NextHop
1	R3
2	R1

With directly connected networks:

NetworkNum	NextHop
1	R3
2	R1
3	Interface 1
4	Interface 0

- If H1 sends to H8, it sends the packet to its default router R1.
- If R2 is R1's default router, R1 forwards the packet to R2.
- R2 sends the packet to next hop R3. R3 delivers packet to H8.

Remarks

- In the example, the forwarding table of R2, with only four entries, gives R2 enough information to reach all 8 hosts in the internetwork.
- The same table would work if each physical network had hundreds of nodes.

➠ Hierarchical addressing (splitting the address into network part and host part) has improved the **scalability** of the network: Forwarding tables only need entries for each network, not for each host.

3.1.5 Address Translation (ARP)

- If a router or host wants to deliver a packet to an IP address in the same physical network, it needs to use the addressing scheme of that particular network.
  - For example, if the network is an Ethernet, it needs to know the 48-bit MAC address of the destination.
- The **Address Resolution Protocol (ARP)** enables each host on a network to build a mapping between IP addresses and link-layer addresses:
  - To determine the MAC address for a particular IP address, a node broadcasts an **ARP query** (containing that IP address) in the network.
  - The node with that IP address responds to the sender of the ARP query with an **ARP response** containing its link-layer address.
  - Nodes maintain an **ARP cache** or **ARP table**.

Classless Routing (CIDR)

- "Classful" IP addresses with network part and host part make routing in the Internet somewhat scalable, but a router still needs to know about **all** networks connected to the Internet, and there are many of them.
- **Solution:** Classless interdomain routing (CIDR, pronounced "cider").
- CIDR uses **aggregation** to minimize the number of routes that a router needs to know.
  - For example, class C network numbers from 192.4.16 to 192.4.31 have the same top 20 bits:
    11000000 00000100 0001
  - If these 16 networks are close to each other, the top 20 bits can be used like a 20-bit network number.

Route Aggregation

- Effectively, CIDR means that network addresses can be of arbitrary length ("classless" addressing).
- Route aggregation can happen repeatedly.
  - For example, if a provider network has assigned 20-bit network addresses

    11000000000001000000

    and 11000000000001000001

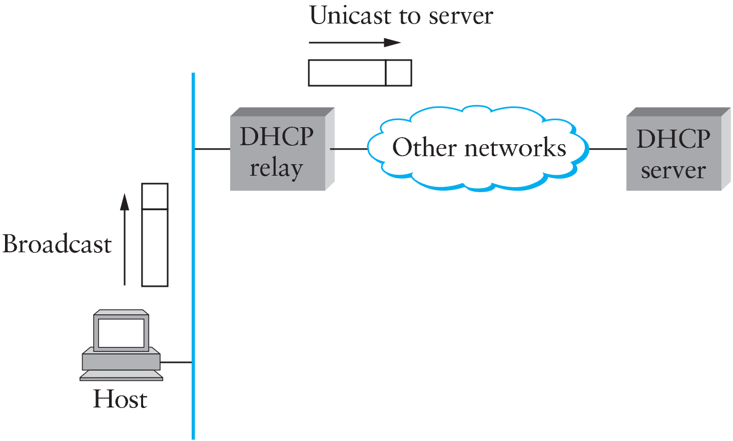
    to two customers, it can advertise a route to the common 19-bit prefix to the rest of the network:

    1100000000000100000
  - All routers in the rest of the network treat the 19-bit prefix as a single network (➠ scalability).
  - With CIDR, routers forward packets according to the "longest match" in their forwarding table.

3.1.6 Host Configuration (DHCP)

- Manual configuration of IP addresses in a local area network is tedious and error-prone.
- Primary automated configuration method: **Dynamic Host Configuration Protocol (DHCP)**
  - A newly booted or attached hosts sends a DHCPDISCOVER message to 255.255.255.255 (IP broadcast in local network).
  - DHCP server responds with IP address assigned to host.
  - Assignment of IP addresses to hosts can be **static** (IP address is always the same for each MAC address) or **dynamic** (server maintains pool of available IP addresses and assigns them dynamically).

DHCP Relay Agents



- To avoid having a DHCP server on every network, **relay agents** can forward DHCPDISCOVER messages to the DHCP server (and replies back to the host).

3.1.7 Error Reporting (ICMP)

- **Internet Control Message Protocol (ICMP)**, a companion protocol of IP, defines error messages sent back to the source host when an IP datagram cannot be processed successfully.
  - Destination host unreachable (e.g. link failure).
  - Reassembly process failed.
  - TTL (time to live) has reached 0. (➠ **traceroute**)
  - IP header checksum failed.
  - ...
- ICMP also defines control messages that a router can send back to a source host, e.g. an ICMP-Redirect that tells the host that there is a better route to the destination.

3.2 Routing

- **Forwarding**: Taking a packet, looking at its destination address, consulting the forwarding table, and sending the packet to the next hop router specified in the table.
- **Routing**: The distributed process by which the forwarding tables at the nodes are built.
- We can distinguish **routing tables**, i.e., tables containing a mapping from network numbers (or prefixes) to next-hop routers, and **forwarding tables**, i.e., tables giving for each network number (or prefix) the outgoing interface and MAC information.
  - Routing tables are produced by the routing algorithm.
  - Forwarding tables are built from the routing tables.

Routing and Forwarding Tables

- **Routing table**:

Network Number	NextHop
10	171.69.245.10

- **Forwarding table**:

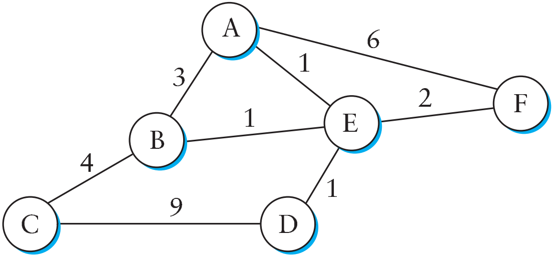
Network Number	Interface	MAC Address
10	if0	8:0:2b:e4:b:1:2

- Routing table and forwarding table contain essentially the same information, but the forwarding table is stored in such a way that the lookup problem can be solved quickly, and it also contains all forwarding information in the form that is needed to do the actual forwarding (interface, and MAC address of next-hop router).

Routing Algorithms

- First, we consider **intradomain** routing, or **interior gateway protocols (IGPs)**.
    - A routing **domain** is an internetwork in which all routers are under the same administrative control (e.g. a university campus or an ISP).
    - Intradomain routing algorithms are suitable in the context of small to midsized networks only (fewer than 100 nodes, in practice).
    - Intradomain routing is complemented by **interdomain routing** between the different routing domains.

3.2.1 Network as a Graph



- Nodes: hosts, switches, routers, networks
- Edges: network links
- Edge costs: indicate desirability of sending traffic over that link
- Routing should find the lowest-cost path between any two nodes.

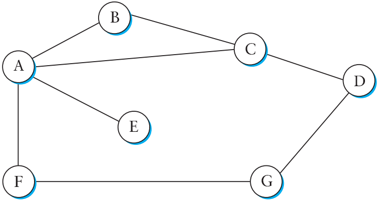
Need for Distributed Algorithms

- The network changes dynamically (node or link failures, addition of new nodes or links, heavily loaded links receive higher costs).
- Therefore, it is not sufficient to compute all shortest paths once and then store the respective information in all nodes.
- In practice, routing protocols provide **distributed, dynamic** ways to maintain shortest paths in a changing network.
- Distributed computation faces many challenges, e.g., two routers may have different ideas about the shortest path to a destination and pass a packet back and forth between them.

3.2.2 Distance Vector Routing (RIP)

- Distance-vector routing is based on the Bellman-Ford algorithm for shortest paths.
- **Basic ideas:**
  - Each node constructs an array (vector) containing the distances to all other nodes, and sends it to its neighbours.
  - If a node learns a shorter path to a destination from a neighbour, it updates its own distance information for that destination.
  - Initially, each node knows the costs of the links to its directly connected neighbours. Links that are down are assigned infinite cost.

Example: Initial Distances



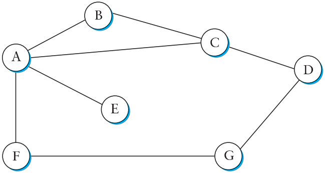
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

- All edge costs in this example are 1.
- Each node only knows the information in one row of the table.

Initial Routing Table

- Node A's initial routing table:

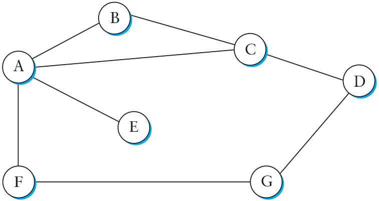
Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	–
E	1	E
F	1	F
G	∞	–



- After receiving distance vectors of its neighbours:

Destination	Cost	NextHop	
B	1	B	
C	1	C	
D	2	C	(updated as C has distance 1 to D)
E	1	E	
F	1	F	
G	2	F	(updated as F has distance 1 to G)

Final Distances



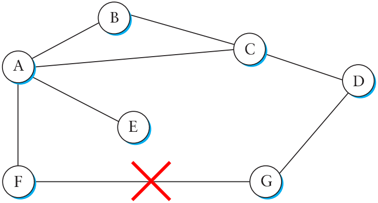
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

- After a few exchanges of information between neighbours, all nodes have consistent routing tables and correct distance information (**convergence**).
- Still, each node knows only one row of the matrix.

Remarks

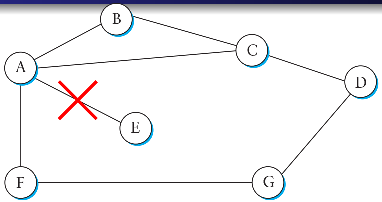
- All nodes achieve a consistent view of the network in the absence of any centralised authority.
- When does a node send distance information to its neighbours?
  - **Periodic update**: Typically, every few seconds or minutes.
  - **Triggered update**: Whenever a node's routing table changes, it sends its updated distance information to its neighbours.
- When a link or node fails, the neighbouring nodes notice it, update their distances, and send updated distances to their neighbours. Normally the system settles down fairly quickly to a new consistent state.

Failure Example



- When the link F–G fails, F detects it, sets its distance to G to ∞, and passes this on to A.
- A knows that its 2-hop path to G is through F, so A also sets its distance to G to ∞.
- With the next routing update from C (which has a 2-hop path to G), A learns that it has a 3-hop path to G (through C).
- When A advertises this to F, node F learns that it has a path to G of cost 4 (through A).

Count-to-Infinity Problem



- A–E fails.
- A advertises distance ∞ to E, but B and C may still advertise distance 2 to E.
- B concludes it has a path of distance 3 to E (via C) and advertises it.
- A thinks it has a path of distance 4 to E (via B) and advertises it.
- C thinks it has a path of distance 5 to E (via A) and advertises it.

➠ Cycle continues until distance values reach ∞. Until then, the network does not stabilise.

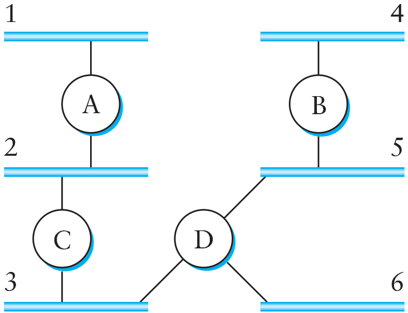
Partial Solutions

- Use a relatively small number as an approximation of infinity, e.g. 16.
- **Split horizon**: When a node sends a routing update to its neighbour, it does not send those routes that it learned from that neighbour.
- **Split horizon with poison**: When a node sends a routing update to its neighbour, it send those routes that it learned from that neighbour as well, but it puts negative information in the route to ensure that the neighbour will not use it (e.g. path cost ∞).
- However, the split horizon techniques only work for routing loops involving two nodes.

Routing Information Protocol (RIP)

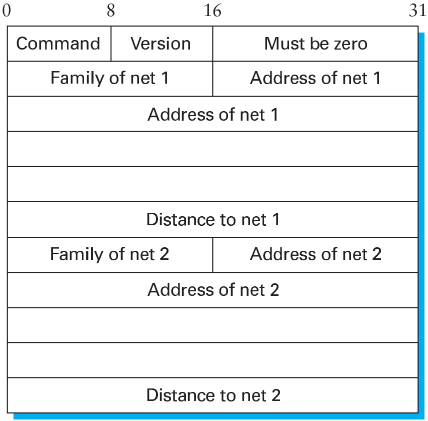
- One of the most widely used routing protocols in IP networks.
- Was distributed with the BSD version of Unix.
- RIP is the canonical example of a routing protocol built on the distance-vector algorithm.
- Routers running RIP actually advertise distances to **networks**. They send periodic updates every 30 seconds (and triggered updates when the routing table changes).
- RIP uses link costs equal to 1, and it uses the number 16 to represent infinity (so a network running RIP must not have a shortest path of more than 15 hops).

Example Network Running RIP



- For example, router C advertises to router A that it can reach:
    - networks 2 and 3 at cost 0
    - networks 5 and 6 at cost 1
    - network 4 at cost 2

RIP Packet Format



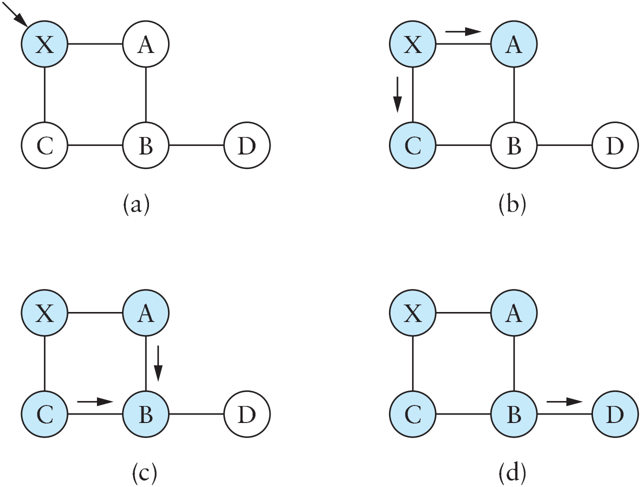
0	8	16	31
Command	Version	Must be zero	
Family of net 1		Address of net 1	
Address of net 1			
Distance to net 1			
Family of net 2		Address of net 2	
Address of net 2			
Distance to net 2			

- Majority of the packet consists of $\langle$network-address,distance$\rangle$ pairs.
- RIP supports multiple address families, not just IP addresses.

3.2.3 Link-State Routing (OSPF)

- Second major class of intradomain routing protocols.
- Idea: Disseminate information about the whole network to every node.
- Two basic mechanisms:
  - Reliable dissemination of link-state information.
  - Calculation of routes from the sum of all the accumulated link-state knowledge.
- Initially, each node knows only the state of the link to each neighbour (up or down) and its cost.
- The information of each node (states and costs of the links to its neighbours) is put into a **link-state packet** (LSP) and **flooded** to all other nodes.

Flooding a LSP



- Each node forwards the LSP to all neighbours except the one from which the LSP was received.

Remarks

- LSPs are generated periodically and in response to topology changes.
- LSPs contain sequence numbers that make it possible to distinguish a new LSP from an older one.
- Once a node has a copy of the LSP from every other node, it can compute a complete map of the network topology.
- From the complete network map, a node can then compute the shortest paths to all other nodes (e.g. using the **Dijkstra** algorithm) and build its routing table.

The OSPF Protocol

- OSPF = Open Shortest Path First
- Most widely used link-state routing protocol.
- Uses **authentication** of routing messages.
  - Protection against misconfigured routers etc.
- Introduces additional hierarchy by allowing a routing domain to be partitioned into **areas**.
  - Reduces amount of information that must be transmitted to and stored in each node.
  - A router does not necessarily need to know how to reach each network in its domain; it may be sufficient to know how to get to the right area.
- **Load balancing**: OSPF allows to distribute traffic evenly among multiple routes of the same cost.

3.2.4 Metrics

- Both distance-vector and link-state routing use link costs (metrics) for shortest path calculations.
- It is **not easy** to determine appropriate link costs.
    - Simplest approach: Assign cost 1 to all links.
        - Used by RIP.
        - Not good because it completely ignores latency, capacity, and load.
    - Original ARPANET routing metric: Number of packets in the queue for the link.
        - Did not work well as it moves packets towards the shortest queue rather than towards the destination.
        - Does not take bandwidth or latency into account.

Second ARPANET Metric

- Second version of ARPANET routing metric ("new routing mechanism"): Compute packet delay as

  $$PacketDelay = (DepartTime - ArrivalTime) + TransmissionTime + Latency$$

  and use average packet delay as link metric.
  - Works reasonably well under light load (where the static values TransmissionTime and Latency dominate the cost).
  - However, bad behaviour under heavy load:
    - Cost of congested link goes up, all traffic moves away from it, link becomes idle and cost goes down, traffic moves back to the link ➠ many links are idle a lot of the time under heavy load.
  - Other problem: Range of link values was too large.

Third ARPANET Metric

- Third version: “revised ARPANET routing mechanism.”
- Major changes:
  - Dynamic range of the metric reduced.
  - Link type taken into account.
  - Variation of metric smoothed over time.
- Properties:
  - Highly loaded link at most 3 times as expensive as when it is idle.
  - Most expensive link at most 7 times as expensive as cheapest link.
  - High-speed satellite link more attractive than low-speed terrestrial link.
  - Link utilisation affects cost only at moderate to high loads.

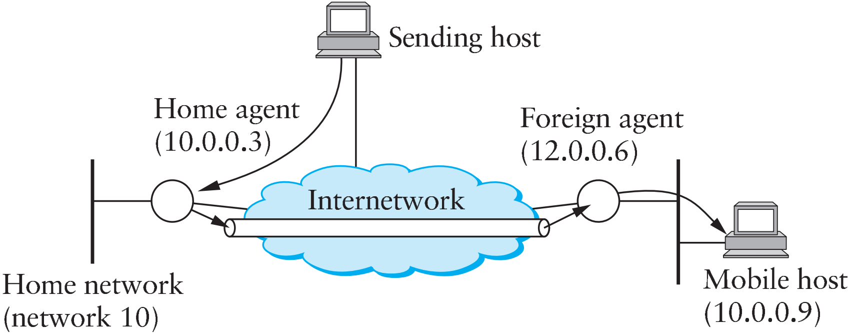
3.2.5 Routing for Mobile Hosts

- A host's IP address consists of network part and host part.
- But: A mobile host may move from network to network.
- A host normally gets a new IP address when it moves to a new network (e.g. using DHCP), but we would like this to be **transparent** to the user and to the applications running on the host and their remote counterparts.
- Procedures to address this problem are referred to as **Mobile IP**.
- Design decision for Mobile IP: Should work without any changes to software of nonmobile hosts or the majority of routers in the Internet.

Mobile IP

- Every mobile node has a **home network** with a router that is the **home agent** of the mobile node.
- The mobile node has a permanent IP address, its **home address**, whose network part is that of the home network.
- When the mobile node is in a foreign network, it registers with a **foreign agent** in that network. The foreign agent contacts the home agent and provides a **care-of address**, usually its own IP address.
- Traffic to the mobile node is
    - sent to its home address,
    - intercepted by the home agent and forwarded (IP tunnel) to the foreign agent, and
    - delivered to the mobile node by the foreign agent.

Illustration of Mobile IP



- The home agent uses ARP messages to associate the home address of the mobile node with its own hardware address ("proxy ARP" technique).
- Thus, all traffic sent to the home address reaches the home agent and can be "tunneled" to the foreign agent.

Remarks

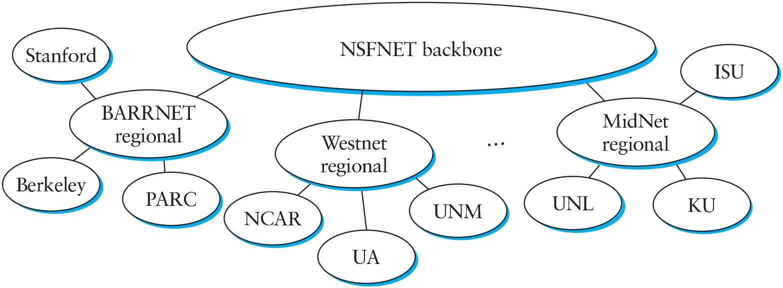
- If the mobile node has acquired a new IP address in the foreign network, it can act as the foreign agent itself.
- Traffic sent from the mobile node to a fixed node can be sent directly: The mobile node uses its home address as source address, and the fixed node's address as destination address.
- If traffic is sent between two mobile hosts, the same mechanism is used in both directions.
- Mobile routing provides **security challenges**: A malicious node wishing to intercept packets for a mobile node could contact the home agent and pretend to be the foreign agent for that node. Therefore, **authentication mechanisms** are required.

Route Optimisation in Mobile IP

- Basic routing mechanism of Mobile IP incurs the "triangle routing problem": Instead of sending traffic directly to the mobile node, it must make a detour through the home agent.
- **Proposed solution**: Inform the sending node about the current care-of address (IP address of foreign agent)!
  - Nodes maintain a **binding cache** of mappings from mobile node addresses to care-of addresses.
  - If a packet for the mobile node arrives at the home agent, the home agent informs the sender about the current care-of address (**binding update** message).
  - Sender then sends packets to foreign agent directly.
  - When the foreign agent receives a packet for a mobile node that is no longer registered with it, it sends a **binding warning** back to the sender.

3.3 Global Internet

- Tree structure of the Internet in 1990:

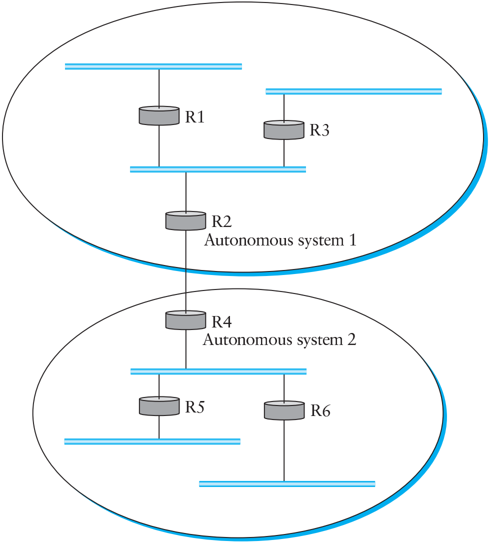


  - End user sites (typically consisting of multiple Ethernets and other networks, connected by routers and bridges).
  - Regional "service provider" networks.
  - National backbone network.

Autonomous Systems

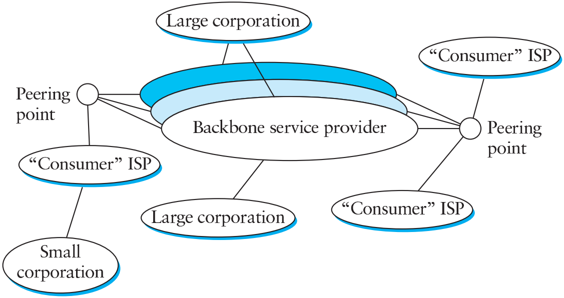
- Each provider and end user is likely to be an administratively independent entity.
- An **Autonomous System (AS)** is a network that is administered independently of other ASs.
- Different providers may have different ideas about the best routing protocol to use within their network, and on how metrics should be assigned to the links.
- Divide the routing problem into two parts:
  - **intradomain** routing (inside a routing domain or AS)
  - **interdomain** routing (between ASs)
- Each AS can run whatever intradomain routing protocol it chooses.

Example with Two ASs



Interdomain Routing Protocols

- First interdomain routing protocol in the recent history of the Internet: **Exterior Gateway Protocol (EGP)**. Had severe limitations, was designed for treelike topology.
- EGP was replaced by the **Border Gateway Protocol (BGP)**, currently in version 4 (BGP-4). Works for arbitrarily interconnected ASs, such as today's multibackbone Internet:



IP Version 6 (IPv6)

- Motivation for a new version of IP: scaling problems caused by the Internet's massive growth; address space being consumed too fast.
- IETF began looking into an expansion of the IP address space in 1991.
- Since longer addresses dictate a change in the IP header, a new version of IP is necessary.
- A new version of IP requires new software in all routers and hosts, so it was felt that one might as well use the new version to fix as many other things in IP as possible.
- New version of IP initially known as **IP Next Generation (IPng)**, but now called **IPv6**.

Wish List Items for IPv6

- Larger address space.
- Support for real-time services.
- Security support.
- Autoconfiguration (i.e., ability of hosts to automatically configure themselves with information such as IP address and domain name).
- Enhanced routing functionality, including support for mobile hosts.

Besides, a transition period during which IPv4 and IPv6 can be used concurrently must be supported.

Note: Many of these features were absent from IPv4 at the time IPv6 was being designed, but support for them has been added to IPv4 in recent years.

IPv6 Addresses

- IPv6 uses 128-bit addresses (➠ 3×10^{38} unique addresses).
- Even based on pessimistic estimates of address assignment efficiency, IPv6 addresses provide over 1500 addresses per square foot of the earth's surface.
- IPv6 address space is subdivided in various ways based on the leading bits.
- Entire functionality of IPv4's main address classes (A, B, and C) is contained inside the 001 prefix (**Aggregatable Global Unicast Addresses**).
- Other prefixes reserved for multicast addresses, link local use addresses, site local use addresses, . . .

IPv6 Addresses (2)

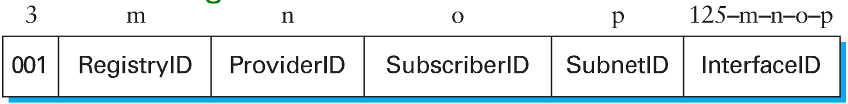
- Part of IPv6 addresses starting 00000000 are reserved for:
  - **IPv4-compatible IPv6 addresses** (32-bit IPv4 address zero-extended to 128 bits)
  - **IPv4-mapped IPv6 addresses** (32-bit IPv4 address prefixed with 2 bytes all 1s and then zero-extended to 128 bits)
- IPv6 addresses are written in the form:

  47CD:1234:4422:AC02:0022:1234:A456:0124
- :: used as short-hand for blocks of 0000 words.
- Special notation for IPv4-mapped IPv6 addresses:

  ::FFFF:128.96.33.81

Aggregatable Global Unicast Addresses

- Address allocation similar to that being deployed with CIDR in IPv4 (so that aggregation can be used effectively).
- Hierarchical assignment of addresses. For example, an IPv6 address **might** look like this:



3	m	n	o	p	125–m–n–o–p
001	RegistryID	ProviderID	SubscriberID	SubnetID	InterfaceID

  - RegistryID: identifier assigned to e.g. a European address registry
  - ProviderID: identifier of service provider
  - SubscriberID: identifier of end user network
  - SubnetID: identifier of physical network

Other Features of IPv6

- **Autoconfiguration**: A host can combine its 48-bit MAC address with the correct prefix (advertised by a router) or with the standard link local use prefix (if it does not need a globally unique address) to obtain an IP address. No need for a DHCP server.
- **Advanced Routing Capabilities**: IPv6 packets can have a routing header that specifies nodes or areas that the packet should visit on its way to the destination.
    - This feature could be used for provider selection on a packet-by-packet basis, for example.

Deployment of IPv6

- IPv6-capable host operating systems are widely available, and router vendors offer varying degrees of IPv6 support.
- However, IPv6 deployment has not begun in any meaningful way.
- **Network address translation (NAT)** has addressed the problem of IPv4 addresses becoming sparse, thus reducing the need for an upgrade to IPv6.
  - NAT allows all nodes in a network to share one (or few) globally unique IPv4 addresses.
- It is not clear when IPv6 deployment will begin in earnest, and what will cause it.
  - Applications that do not work well with NAT could be driving factor (multiplayer gaming, IP telephony, etc.).