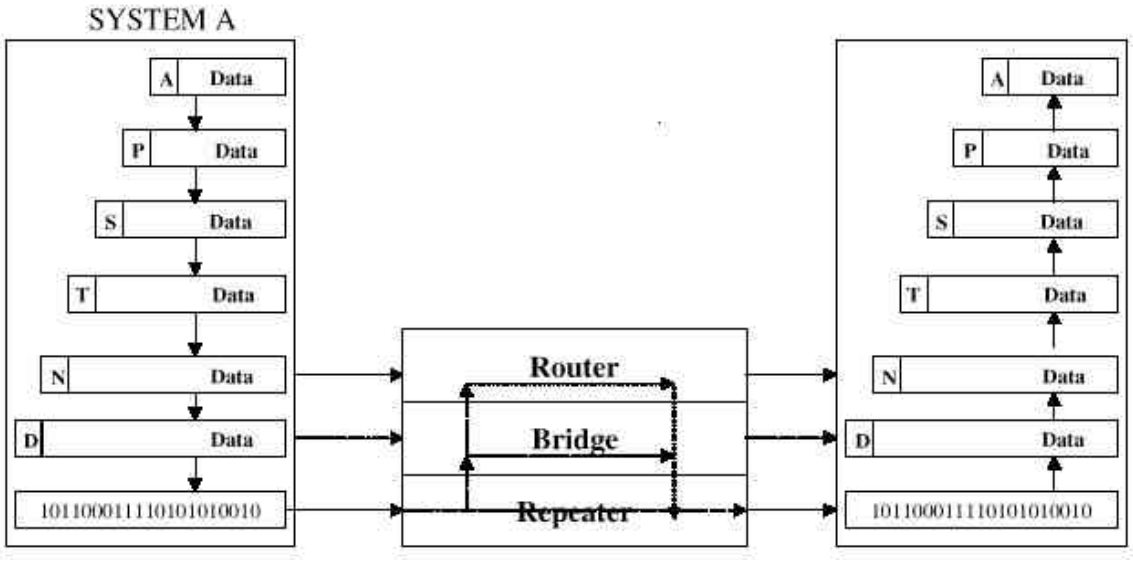
1. Repeaters, Bridges and Routers

The figure shows the 7 layers of the OSI model. This model was discussed in detail in the Datacomms 2 course. We will revise some of the main points made and describe the concept of Bridging and Routing in relation to this model.

Each layer operates independently of the others using a method referred to as encapsulation. At the sending device each layer receiving data from the layer above will process the data, add its own protocol header and transfer the data block to the layer below. The layer below will simply treat the data as a data block, it will not try to understand its meaning. The block will be processed by the layer, which adds its own protocol header and then passes the larger data block to the layer below. At the receiving device the reverse happens. When the data arrives, the first layer processes its peer header and then passes the data to the layer above which carries out the same action. Ultimately, the application data originally sent by the sending device will arrive at the receiving application.

Routers operate at the network layer. They connect networks into internetworks that are physically unified, but in which each network retains its identity as a separate network environment.

Bridges operate at the Data link layer. They connect network environments into logical and physical single internetworks.
Repeaters operate at the Physical layer. They receive transmissions (bits) on a LAN segment and regenerate the bits to boost a degraded signal and extend the length of the LAN segment.



To understand one of the key differences between internetworking products it is essential to appreciate what a collision domain and a broadcast domain is and the effect that each of the products has on these domains.

Collision Domain - If two devices within the domain attempt to transmit simultaneously the packets will collide and re-transmission will occur.

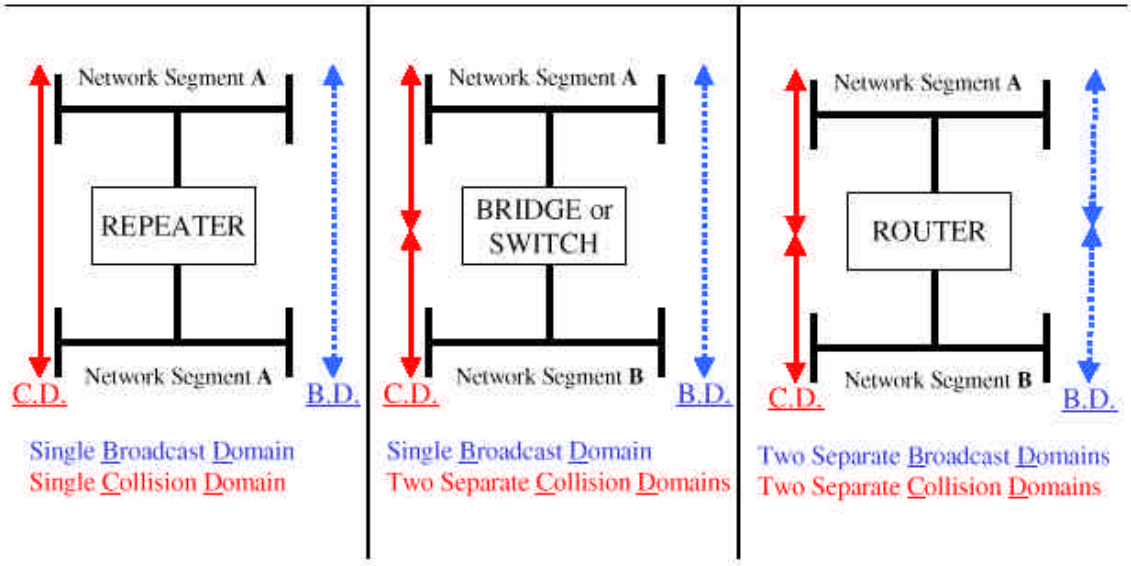
Broadcast Domain - If a device sends out a Network layer broadcast, for example, ARP request, it will be received by all devices within the same broadcast domain.

Repeaters only regenerate the signal. They do not in anyway reduce network collisions or broadcast traffic.

Bridges (and Switches) reduce the number of collision on the network by breaking the network into smaller segments. Two devices on either side of a bridge can put traffic on the LAN simultaneously and they will never collide with each other.

(Note: A LAN switch is effectively a high-speed bridge and the details in this chapter apply to both devices)

Routers like bridges reduce the number of collisions. In addition to this they stop network broadcast traffic, thus reducing the amount of traffic on each segment.



Types of Bridges

A bridge is an electronic device that connects two LAN segments. A bridge forwards complete, correct frames from one segment to another.
A typical bridge consists of a conventional computer with a CPU, memory, and two network interfaces. It is dedicated to a single task and does not run application software.

Bridges are used to span longer distances in networks. For example, a corporation may need a network that allows computers in one building to communicate with computers in another. If the two buildings are separated by a significant distance or if the buildings are large, a single LAN will not suffice to reach both buildings. On the other hand, using optical fibre would be very costly.

Several kinds of bridges have emerged as important. These are:

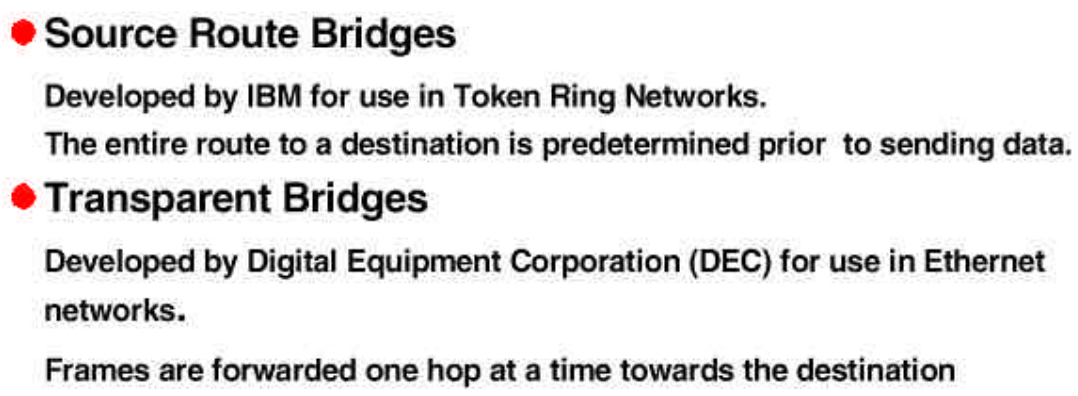
- Source-Route Bridges
- Transparent Bridge

Source-Route Bridging

Source-Route Bridging (SRB) was developed by IBM for use in Token Ring networks. With SRB, the source places the complete source-to-destination route in the frame header of all inter-LAN frames. To discover a route to the destination, the source sends an explorer frame to determine where the destination is located.

Transparent Bridging

Transparent bridging was developed by Digital Equipment Corporation (DEC). It is most often found in Ethernet networks, in which bridges pass frames along one hop at a time, based on tables associating end nodes with bridge interfaces. Transparent bridges are designed to enable frames to move back and forth between network segments running the same MAC layer protocols. It is referred to as transparent bridging because the presence of the bridges is transparent to other network devices. The bridges do not alter the data frame and the address of the bridge is never the source or destination of a frame.



Source-Route Bridging

Source route bridging is used primarily in Token Ring networks. Source routing assumes that the sender of each frame knows whether or not the destination is on its own LAN. When sending a frame to a different LAN, the source sets the high-order bit of the source address to 1, to mark it. Furthermore, the exact path that the frame will follow is included in the frame header.

The path is constructed as follows. Each LAN has a unique 12-bit number, and each bridge has a unique 4-bit number that identifies it in the context of its LANs. A route is therefore a sequence of bridge, LAN, bridge, LAN, numbers.

A source route bridge is only interested in those frames with the high-order bit of the destination set to 1. For each such frame it sees, it scans the route looking for the number of the LAN on which the frame arrived. If this LAN number is followed by its own bridge number, the bridge forwards the frame onto the LAN whose number follows its bridge number in the route. If the incoming LAN number is followed by the number of some other bridge, it does

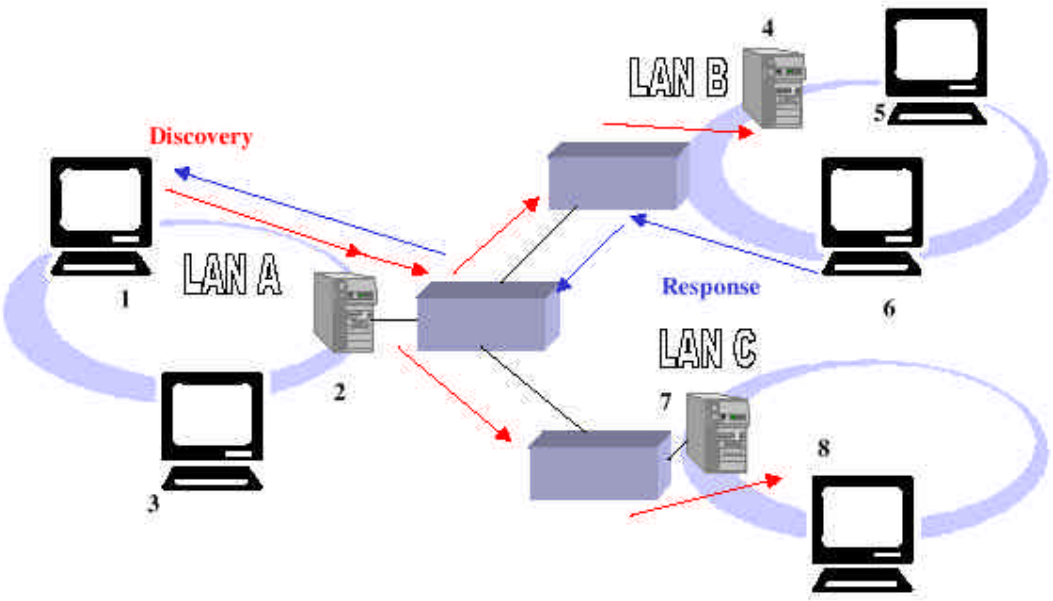
not forward the frame.
This algorithm lends itself to three possible implementations. These three implementations vary in cost and performance.

1. Software: the bridge runs in promiscuous mode, copying all frames to its memory to see if they have the high-order destination bit set to 1. This implementation requires a very fast CPU.

2. Hybrid: the bridge's LAN interface inspects the high-order destination bit and only accepts frames with the bit set. This interface is easy to build into hardware and greatly reduces the number of frames the bridge must inspect.

3. Hardware: the bridge's LAN interface not only checks the high-order destination bit, but it also scans the route to see if this bridge must forward the frame. Only frames that must actually be forwarded are given to the bridge. This implementation requires the most complex hardware, but wastes no CPU time because all irrelevant frames are screened out. This implementation requires a special VLSI chip, but offloads much of the processing from the bridge to the chip, so that a slower CPU can be used, or alternatively, the bridge can handle more LANs.

Every machine in the internetwork knows, or can find, the best path to every other machine. How these routes are discovered is an important part of the source routing algorithm. The basic idea is that if a destination is unknown, the source issues a broadcast frame asking where it is. The discovery frame is forwarded by every bridge so that it reaches every LAN on the internetwork. When the reply comes back, every bridge on its route records it's identity in the reply, so that the original sender can see the exact route taken, and ultimately choose the best route. Once a host has discovered a route to a certain destination, it stores the route in the cache.



Transparent Bridging Operation

There are three processes involved in transparent bridging operation. These are:

- Learning
- Forwarding
- Filtering

Learning

When a transparent bridge is first turned on, it knows nothing about the network topology. It learns which devices can be reached on each of its interfaces by monitoring the source MAC address of all incoming frames.

It maintains a database of these learned Media Access Control (MAC) addresses and their associated interfaces in a table. The bridge updates this table every time a device sends a frame, and deletes entries of devices not heard from within a specified time period.

This learning capability allows new devices to be added to the network without reconfiguring the bridge.

Forwarding

If a bridge knows where a destination address is, it forwards frames out the associated interface. If the bridge does not know where the destination address is, it forwards the frame out every interface. This is called flooding.

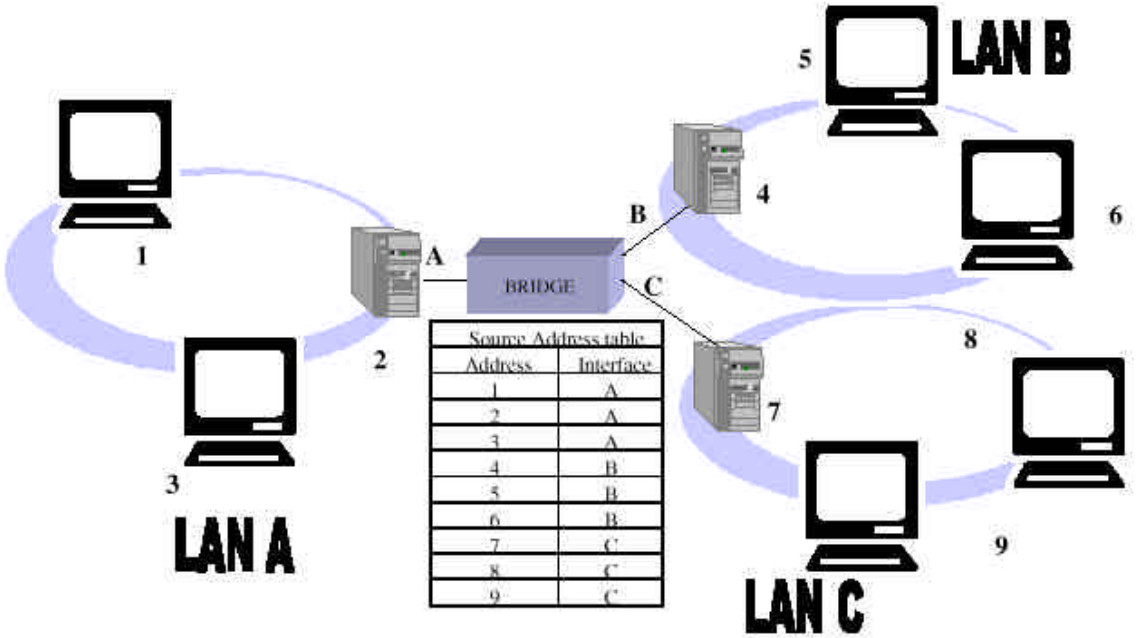
A bridge learns addresses and forwards traffic as follows: Assume that the source and destination addresses are located on different bridged networks, and neither address is known to the bridge. The bridge notes the source address and updates its tables. It forwards the frame out all interfaces, except the one where it was received. If a reply comes back, the bridge examines the source address, which was the original target address, and adds the entry to its table.

The bridge forwards all subsequent communication between the devices.

Filtering

Typically, about 80 per cent of the frames transmitted on a typical workgroup or department LAN are destined for stations on the local LAN. Bridges make a simple ‘forward’ or ‘don’t forward’ decision on each frame they receive from the LAN. If a frame’s destination address is on the same LAN segment as it’s originating address, it is filtered out and not forwarded across the bridge.

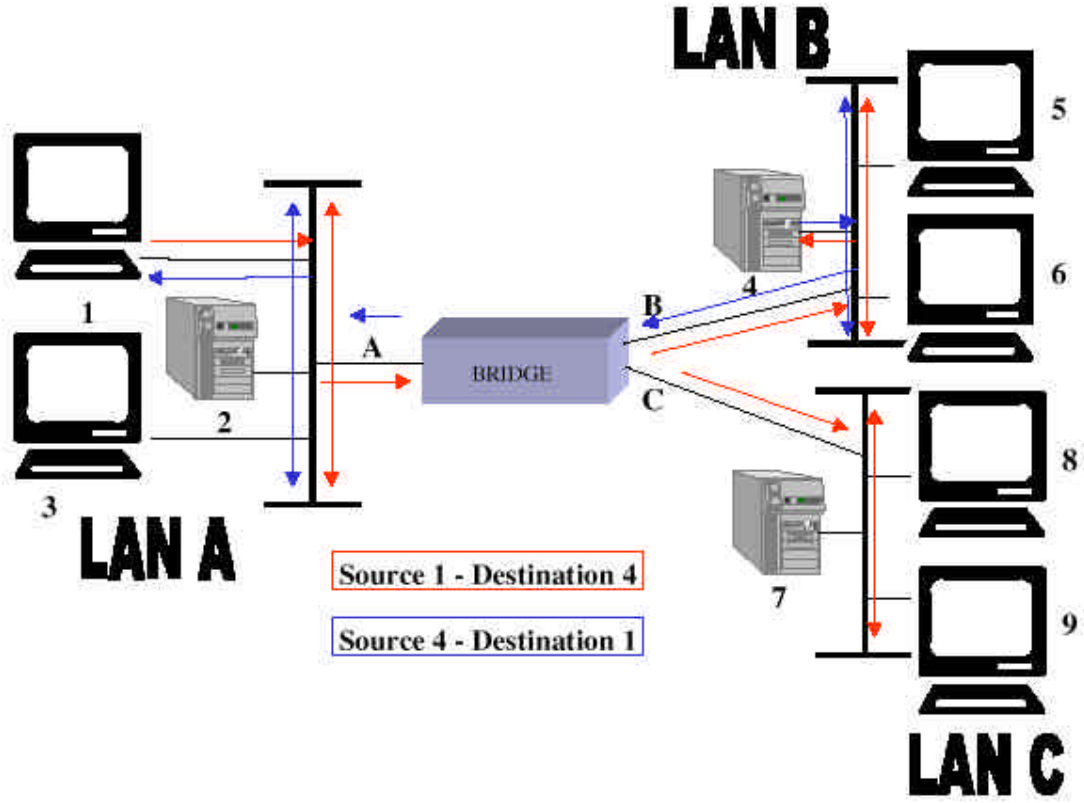
Bridges can filter frames based on any link layer field. For example, a bridge can be configured to reject all frames from a particular network. Unnecessary broadcast and multicast frames can also be filtered in this way. Data-link information often includes a reference to an upper-layer protocol, and bridges can usually filter based on this parameter too.



Transparent Bridge Operation-Example

Device 1 on LAN A addresses a packet to device 4 on LAN B. The bridge receives this packet on Interface A and floods it out every other interface. The bridge now knows that address 1 is out interface A. The packet is received by device 4 and it replies with a packet which has a destination 1 and source 4.

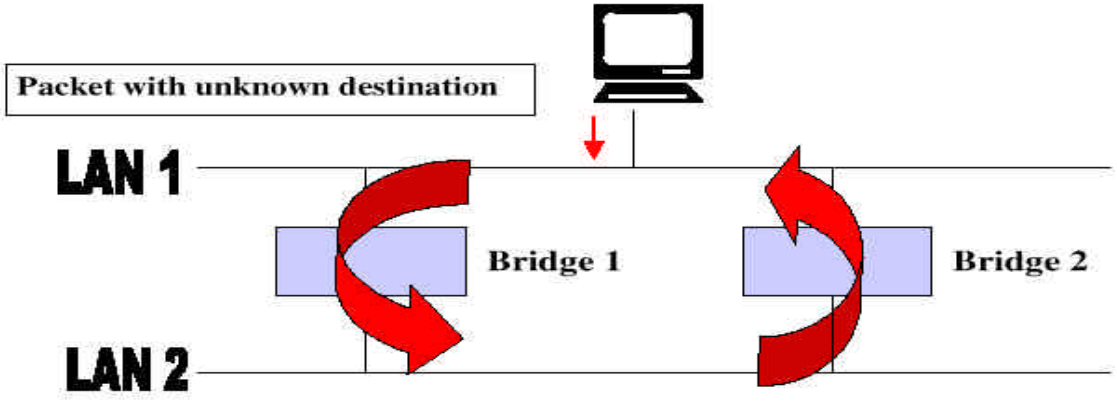
The bridge receives this packet on interface B, so it now knows that address 4 is out interface B. The bridge forwards the packet out interface A only, as it already knows where device 1 is. In this way, the bridge has built up and stored two entries in its source address table.



Bridging Loops

To increase reliability it is common practice to use two or more bridges in parallel between pairs of LANs. This arrangement, however, also introduces some additional problems because it causes loops in the topology.

For example, if a packet with an unknown destination arrives at bridge 1 from LAN 1, it forwards it onto LAN 2. Bridge 2 now sees this packet on LAN 2 and, since the destination is still unknown, it forwards it onto LAN 1. Once again, bridge 1 sees the packet on LAN 1 and forwards it onto LAN 2. This cycle could go on forever, using up the bandwidth and blocking the transmission of other packets on both segments.

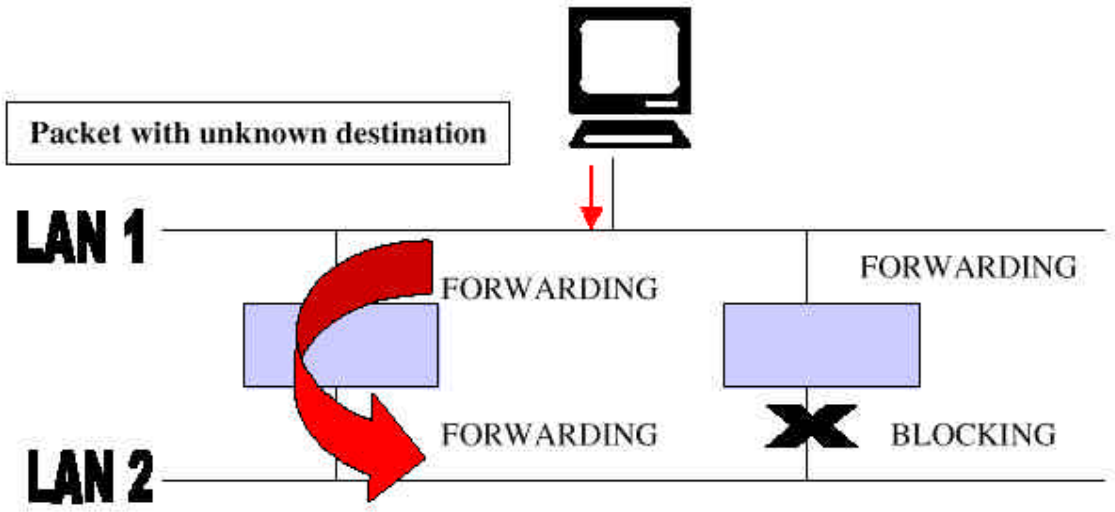


Preventing Loops

The Spanning Tree Protocol, sometimes referred to as the Spanning Tree Algorithm (STA), solves the problems associated with bridge loops. It allows redundant paths and ensures a loop-free topology by means of a bridge-to-bridge protocol. It creates this loop-free topology by blocking duplicate paths between network segments and automatically activating backup paths if a link segment or bridge fails.

The STA creates a set of device-to-device paths through the network, such that there is only one active or 'primary' path between any two devices. All paths not selected by the STA are temporarily disabled.

STA allows participating bridges to reactivate blocked paths if an existing primary path fails. With this feature, the STA allows networks to recover quickly and automatically if a network device, such as a bridge or a section of networking cabling fails.



Spanning Tree Protocol

The STP elects the bridge with the lowest priority to be the root bridge. This priority can be configured by a network administrator. If it is not, then the bridge with the lowest value identifier (based on the MAC address plus a priority value) becomes the root by default.

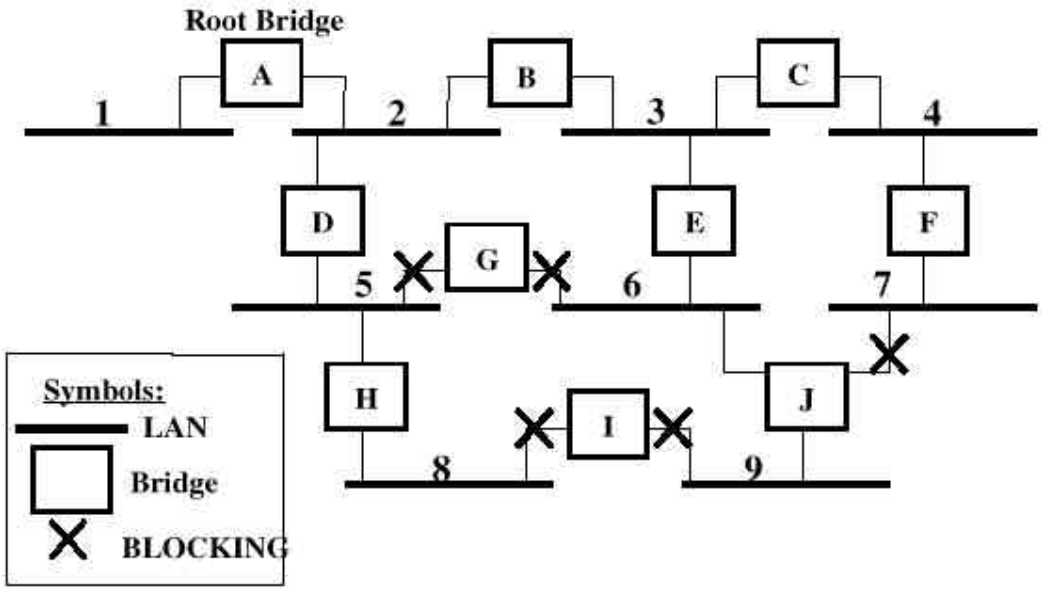
Every other bridge selects the lowest-cost path to the root bridge. Interface costs can be altered by a network administrator in order to select a preferred route.

All interfaces on these paths forward traffic. All interfaces not on these paths block traffic. This ensures that a unique path is established from every LAN to root. The algorithm runs continuously to detect topology changes and update the tree.

Initially, all bridges consider themselves to be the root bridge. Each bridge broadcasts a Bridge Protocol Data Unit (BPDU) on each of its LANs that asserts this fact. On any given LAN, only one claimant has the lowest-valued identifier and maintains this belief. Over time, as BPDUs propagate, the identity

of the lowest-valued bridge identifier throughout the internet becomes known to all bridges. The root bridge regularly broadcasts the fact that it is the root bridge on all the LANs to which it is attached. This allows the bridges on those LANs to determine their root port and the fact that they are directly connected to the root bridge. Each of these bridges in turn broadcast a BPDU on the other LANs to which it is attached (all LANs except the one on its root port), indicating that it is one hop away from the root bridge. This activity is propagated throughout the internet. Every time a bridge receives a BPDU, it transmits BPDUs, indicating the identity of the root bridge and the number of hops to reach the root bridge.

On any LAN, the bridge claiming to be the one closest to the root becomes the designated bridge.



LAN Switches

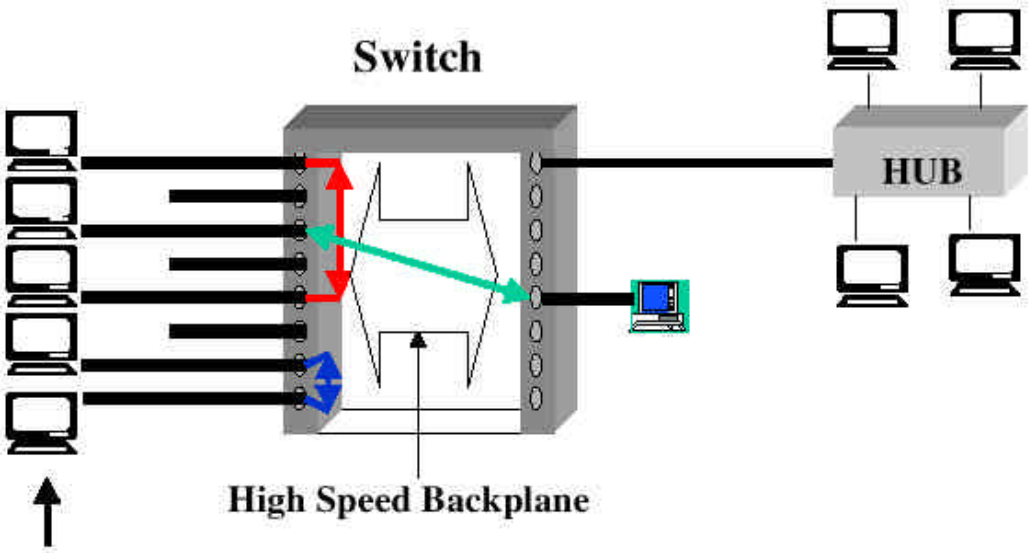
A LAN switch is a network device containing a high-speed backplane (> 1 Gbit/s) and room for a number of plug-in cards. LAN switches are really high capacity bridges and are therefore similar in functionality. Each card typically contains 8 or 16 connectors. Usually, each connector has a 10Base-T twisted pair to a single network device, for example, a file server.
When a device sends a frame, it first arrives at a plug-in card on the switch.

This card checks to see if it is destined for one of the other devices connected to the same card. If so, the frame is copied there. If not, the frame is sent over the high-speed backplane to the destination device card.

Each input port of the plug-in card is buffered, so incoming frames are stored in the card's on-board RAM as they arrive. This design allows all input ports to receive (and transmit) frames at the same time, for parallel full-duplex operation. Each port is a separate collision domain, so collisions do not occur, if only a single device is connected to the port.

It is possible to connect an Ethernet hub to a port on the switch, as both use standard Ethernet frames. Frames arriving at the switch from the hub are

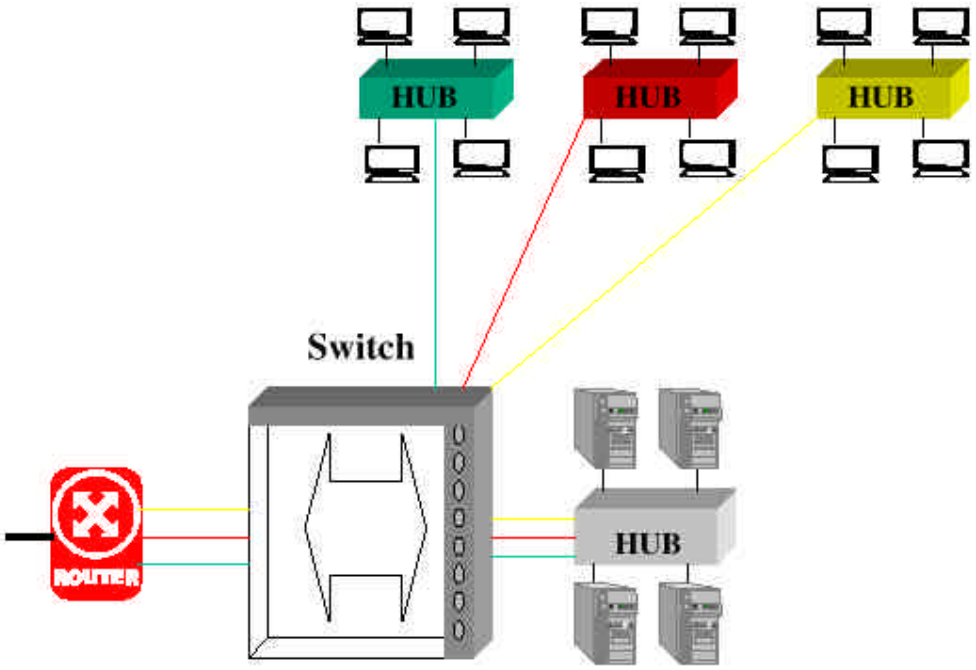
treated there like any other incoming frames, that is, they are switched to the correct output line over the high-speed backplane. If all ports are connected to hubs rather than individual stations, the switch is then a bridge.



VLANs

VLANs allow PCs, workstations, and other resources, including printers and file servers, to be organized into logical, broadcast domains so that only devices within the same domain can communicate with each other. VLANs allow users to implement configurations on their network using one of two schemes: IEEE 802.1Q, including GVRP, which enables the auto-learning of VLANs, or 3Com's VLT. Both methods allow for the configuration of VLANs based on ports and/or MAC addresses for maximum flexibility and security. For 802.1Q VLANs, a port on a switch can be assigned to a VLAN; all other switches learn about that VLAN when the switches automatically communicate that knowledge via the GVRP protocol.

Switches supporting both VLAN schemes can be used to provide seamless migration from VLT to IEEE 802.1Q environments that preserve investment in current LAN developments and equipment.



IEEE Standard 802

The IEEE has produced several standards for LANs. These standards, collectively known as IEEE 802, include CSMA/CD, Token Bus and Token Ring. The various standards differ at the physical layer and MAC sublayer but are comparable at the data link layer. The standards are divided into parts, each published as a separate book.

IEEE Standard 802.3 and Ethernet

The IEEE 802.3 standard is for a CSMA/CD LAN. When a station wants to transmit, it listens to the cable. If the cable is busy, the station waits until it is idle; otherwise, it transmits immediately. If two or more stations begin to transmit simultaneously on an idle cable, they collide. All collision stations then terminate their transmission, wait a random time, and repeat the whole process all over again. The 802.3 standard specifies several physical media, such as coaxial cable for 10Base5 (thick ethernet), 10Base2 (thin coaxial cable), 10Base-T (twisted pair) and 10Base-F (fibre). The 803.2 standard also specifies the 802.3 MAC sublayer protocol and the standards for a switched 802.3 LAN.

How Routers Operate

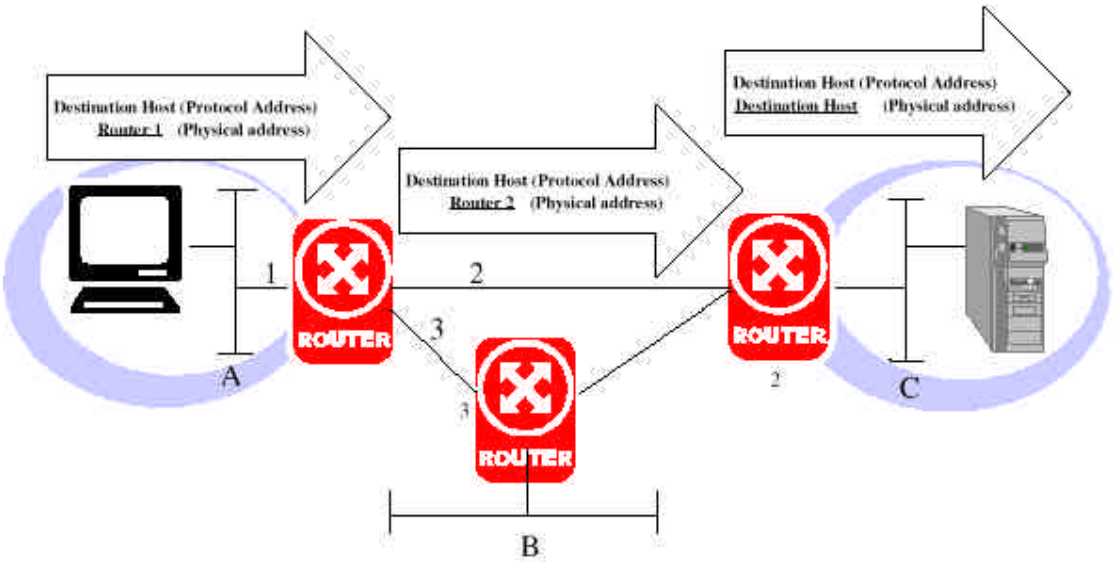
Routers interconnect the various network segments making up the Internet. A router receives an IP packet on one of its interfaces, and forwards the packet out another of its interfaces (or possibly more than one if the packet is a multicast packet), in accordance with the contents of the IP header. As the packet is forwarded hop by hop the packet's network-layer header, the IP header, remains relatively unchanged. However, the data link headers and physical transmission schemes may change radically at each hop in order to match the changing media types.

We will now examine what happens when a router receives a packet from one of its attached Ethernet segments. The router's Ethernet adapter indicates the size of the packet received. The router first looks at the packet's data link header, which in this case is an Ethernet header. If the Ethernet type is set to #0800, indicating an IP packet, the Ethernet header is stripped from the packet, and the IP header is examined. Before discarding the Ethernet header, the router notes the length of the Ethernet packet and whether the packet has been multicast or broadcast on the Ethernet segment by checking a particular bit in the destination MAC address. In some cases routers will refuse to forward data link multicasts or broadcasts.

The router then verifies the contents of the IP header by checking the Version, Internet Header Length (IHL), Length, and Header Checksum fields. The version must be equal to 4. The IHL must be greater than or equal to the minimum IP header size (five 32-bit words). The length of the IP packet expressed in bytes, must be also larger than the minimum header size. In addition, the router should check that the entire packet has been received, by checking the IP length against the size of the received Ethernet packet. Finally, to verify that none of the fields of the header have been corrupted, the 16-bit ones-complement checksum of the entire IP header is calculated and verified. If any of these basic checks fail, the packet is deemed so malformed that it is discarded without even sending an error indication back to the packet's originator.

Next, the router verifies that the Time To Live (TTL) field is greater than 1.The purpose of the TTL field is to make sure that packets do not circulate forever when there are routing loops. Each router decrements the TTL field on the way to a destination. When the TTL field is decremented to 0, the packet is discarded, and an Internet Control Message Protocol (ICMP) TTL Exceeded message is sent back to the host. On decrementing the TTL, the router must adjust the packet's Header Checksum.

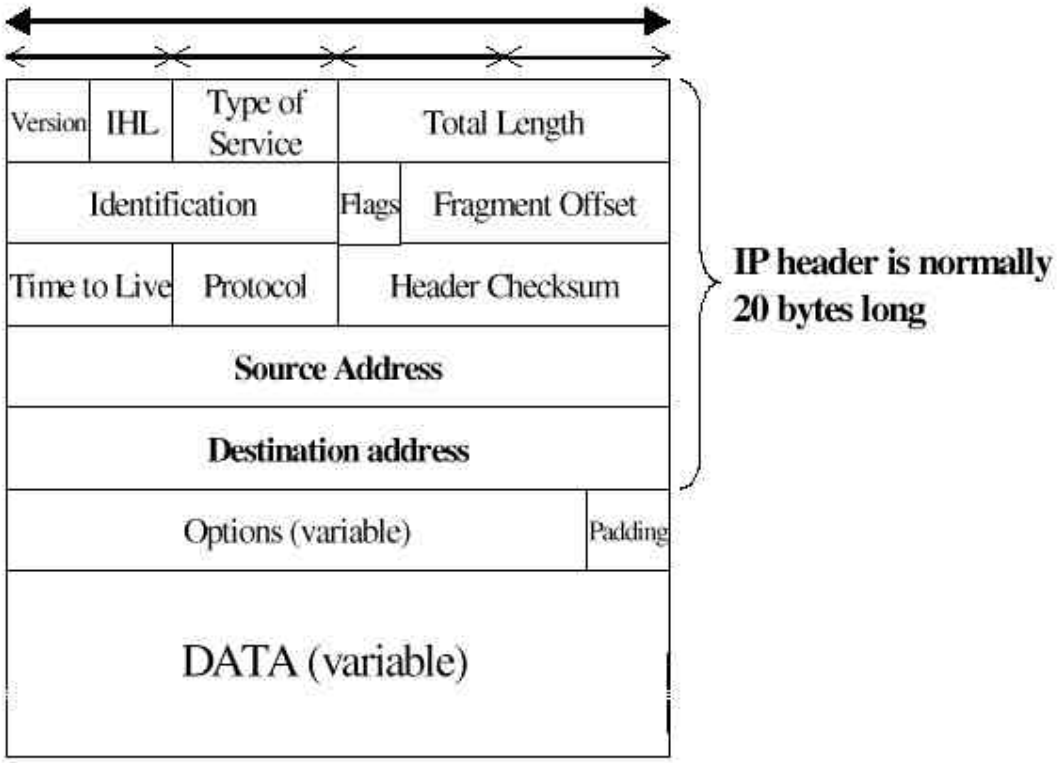
The router then looks at the destination IP address. The destination IP address is used as a key for the routing table lookup. The best matching routing table entry is returned, indicating whether or not to forward the packet. If the packet is to be forwarded, this entry also indicates the interface to forward the packet out of and the IP address of the next IP router.



If the packet is too large to be sent out on the outgoing interface in one piece, that is, its length is greater than the outgoing interface's Maximum Transmission Unit (MTU), the router attempts to split the packet into smaller pieces, called fragments. Fragmentation may affect performance adversely.

Hosts may wish to prevent fragmentation by setting the Don't Fragment (DF) bit in the Fragmentation field. In this case, the router drops the packet and sends an ICMP Destination Unreachable message back to the host. The host uses this message to calculate the minimum MTU along the packet's path, which is used to size future packets.

The router then prepares the appropriate data-link header for the outgoing interface. The IP address of the next hop is converted to a data-link address, usually using ARP or a variant of ARP, such as Inverse ARP. The router then sends the packet to the next hop, where the process is repeated.



We have described how a router forwards an IP packet. However, to start with, an IP packet sent from a host in a network with a destination in another must find a router to send a packet. There are two ways this is done in an Ethernet LAN. These are:

- Using a default gateway
- Using proxy ARP

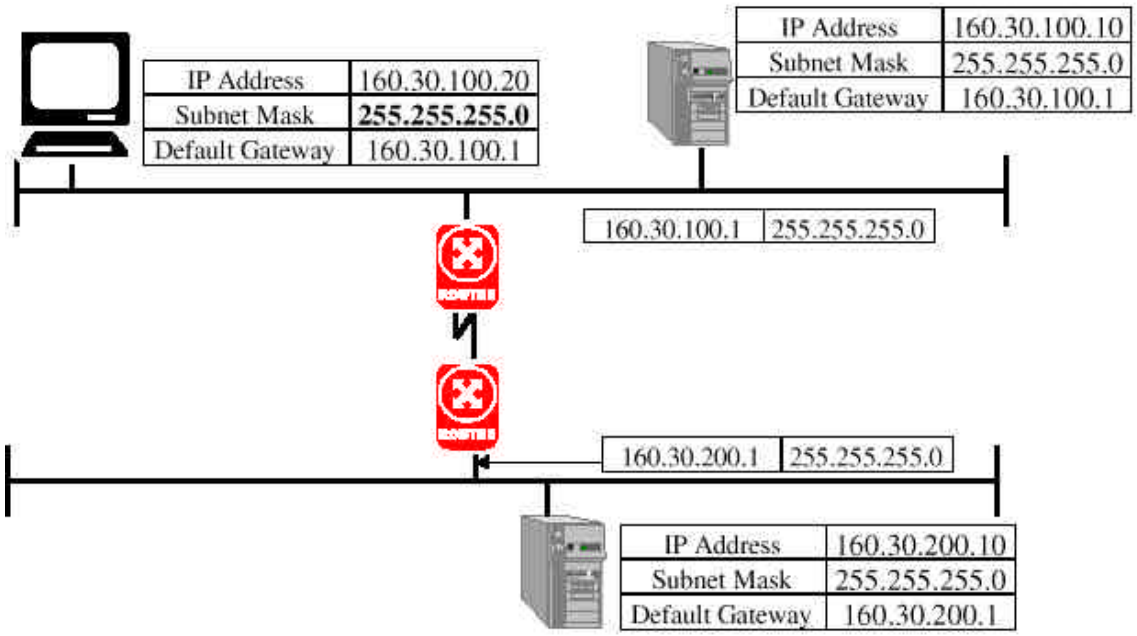
These are described in the following paragraphs.

Using Default Gateway

When a host sends a packet, it must determine the next hop. A host that has one network connection, such as an Ethernet interface, has an IP address assigned to it. The first test that the host performs is to determine whether the packet's destination address belongs to the same subnet. A logical AND is performed with the subnet mask and the destination IP address and compared to the result of a logical AND between the subnet mask and the host's own IP address. If the result is different, the destination is remote and the next hop's address is of a router on the path to this remote location. The host is configured with the IP address of the next hop router, that is, the default gateway.

The host must now find the hardware address of the default gateway. The host broadcasts an ARP request packet over the Ethernet. This is received by all stations. The default gateway recognises the IP address and sends back an ARP reply.

The hosts keep the result of the translation in their cache memories. If the host has to send more packets to the same destination it simply looks into the cache memory and copy the 48-bit hardware address without having to resort to ARP. Requests and responses are identified by the operation code (resp 1 and 2).



Using Proxy ARP

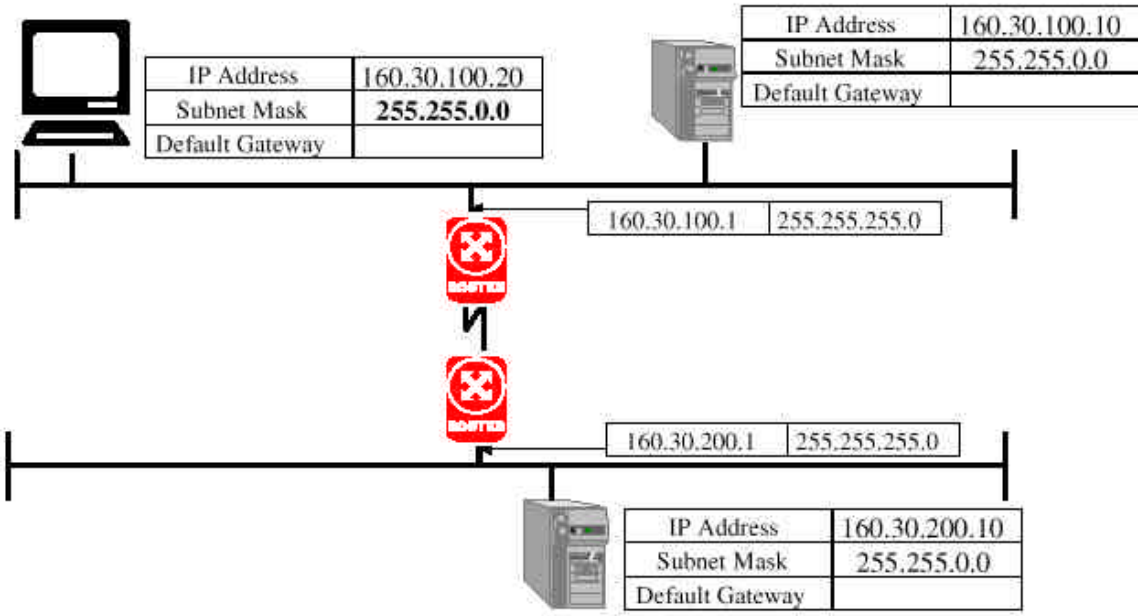
In the example in the following figure the PC and servers are configured with the class B default subnet mask, (255.255.0.0). The routers are configured with the customised mask (255.255.255.0).

If the PC wants to send a packet to the server on the remote network, it compares the destination's network ID and subnet, with its own network ID and subnet. The result implies that they are on the same network, so that the PC tries to send the packet directly using ARP.

Routers do not normally propagate broadcasts, so the actual ARP broadcast does not go beyond the senders network.

However, routers can run a protocol called Proxy ARP. When a router running Proxy ARP receives an ARP request, it reads the packet and applies the subnet mask for the sender's subnet to the requested destination IP address. This gives it the network ID which it compares with its routing tables in order to find a match.

In the example, the router determines that it knows a viable route to get the packet to the subnet of the destination. The router then replies to the ARP request exactly as if the router were itself the destination device. The only difference is that the hardware address returned in the ARP reply is the address of the router port connected to the source network. The source device and router now enter each other's IP/Hardware address pair in their ARP cache and the first data packet can be sent.



All TCP/IP routing protocols have ways of discovering the reachable IP address prefixes and, for each prefix, the next-hop router to use to forward data traffic to the prefix. As the network changes - leased lines fail, new leased lines are provisioned, routers crash, and so on - the routing protocols continually re-evaluate prefix reachability and information about the next hop to use for each prefix. The process of finding the new next hop after the network changes is called convergence. Routing protocols that find the new next hop quickly, that is, protocols having a short convergence time, are preferred.

A router's routing table instructs the router how to forward packets. There is a separate routing table entry for each address prefix that the router knows.

Entries in the routing table are also commonly known as routes. If a packet's IP destination falls into the range of addresses described by a particular routing table entry's prefix, we say that the entry is a match.

Many routers have a default route to external destinations in their routing table, that is, destinations that are not within the routing domain. The default route matches every destination, although it is overwritten by all the more specific prefixes.

There are two types of routing, which are:

- Dynamic routing
- Static routing

Static Routing

Routing table entries can be configured by a network operator. This is called static routing. For example, to install a route to subnet 192.168.10.0, a network operator may type the command:

```
ADD IP ROUTE ENTRY 192.168.10.0 255.255.255.0 192.168.12.1
1
```

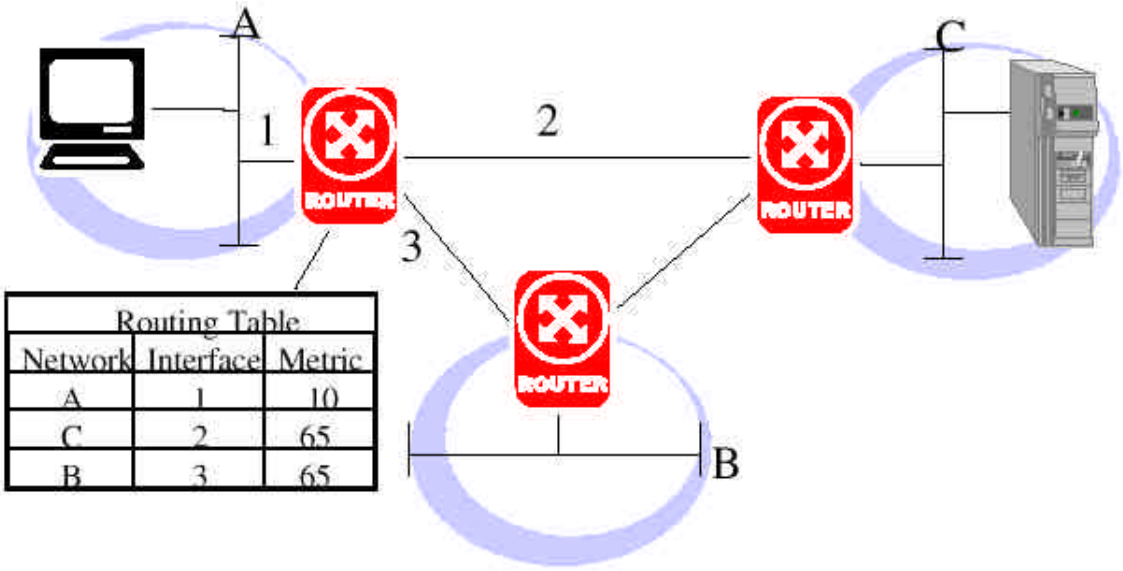
Where 192.168.10.0 is the destination prefix (class C network), 255.255.255.0 is the network mask, and 192.168.12.1 is the IP address of the next hop router,

and 1 is the metric associated with that path. One main disadvantage of static routing is, if a link fails, an alternative link has to be configured manually by the network administrator.

Dynamic Routing

Dymanic routing adjusts in real time to network changes by analysing routing update messages. There are two main processes involved in dynamic routing. These are:

- Information distribution, where each router sends and receives routing information within the internetwork using a routing protocol such as, BGP, OSPF and RIP.
- Route calculation, where each router calculates the best path to each destination using an algorithm and the information received using the routing protocols.



Routing Algorithms

There are two types of algorithms used in routing. These are:

- Distance vector algorithms
- Link state algorithms

Distance Vector Protocols

In a Distance Vector Protocol, the routers cooperate in performing a distributed computation. Distance Vector algorithms calculate the best path to each destination separately, usually trying to find a path that minimises a simple metric, such as the number of hop counts to the destination. Each route has its current best path to a destination and sends this information to its neighbours in routing updates. The router's neighbours also notify the router of their path choices. The router, seeing the paths being used by all of its neighbours, may find a better (that is, low cost) path to a particular destination through one of its neighbours. If so, the router updates its next hop and cost for the destination and notifies its neighbours of its new choice of route, and the procedure iterates. After some iterations, the choice of router stabilises, with each router having found the best path to the destination.

The main advantage of Distance Vector algorithms is their simplicity. The Internet's Routing Information Protocol (RIP) is a good example of a Distance Vector Protocol which uses the Bellmann Ford algorithm.

Link State Algorithms

Link State routing algorithms employ a replicated distributed database approach. Each router contributes pieces of information to this database by describing the router's local environment: the set of active links to local IP networks and neighbouring routers, with each link assigned a cost. Instead of advertising a list of distances to each known destination, a router running link-state algorithm advertises the states of its local network links. These link state advertisements are then distributed to all routers. The end result is that all routers obtain the same database of collected advertisements, together describing the current map of the network. From the network map, each router runs a shortest-path calculation, typically the Dijkstra algorithm. The shortest path in the network is assigned as the sum of the costs of the links comprising the path.

Link-state algorithms are considered to have good convergence properties. When the network changes, new routes are found quickly and with a minimum of routing protocol overhead. Link-state routing protocols are more complicated to specify than are Distance Vector Protocols, as you can tell by comparing the size of the OSPF and RIP specifications.

Routing Metrics

Metrics are used by routing algorithms to select the best route. Sophisticated routing algorithms can use a combination of the following metrics:

- Path length is the sum of the interface costs associated with each network link. Hop count specifies the number of passes through internetworking devices (such as routers) that a packet must take from a source to a destination.
- Reliability is usually assigned to network links by network administrators. The values assigned are based on how frequently the network link goes down and how long it typically takes to be repaired.
- Delay refers to the length of time it takes to move a packet from source to destination through an internetwork. It is dependent on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, network congestion on all intermediate network links, and the physical distance to be travelled.
- Bandwidth refers to the available traffic capacity of a link.
- Load refers to the degree to which a network resource (such as a router) is busy, for example, its CPU utilisation and the number of packets processed per second.
- Communications cost is the actual financial cost associated with a particular route. A network administrator may configure routers so that traffic uses a slower link, if it is cheaper to do so.

Dijkstra Algorithm

The Dijkstra algorithm involves using the information in the link-state database to calculate the routing tables.

The Dijkstra algorithm is a simple algorithm that efficiently calculates all the shortest paths to all destinations at once. The algorithm incrementally calculates a tree of shortest paths. It begins with the calculating router adding itself to the tree. All of the router's neighbours are then added to a candidate list, with costs equal to costs of the links from the router to the neighbours. The router on the candidate list with the smallest cost is then added to the shortest-path tree, and that router's neighbours are then examined for inclusion in the candidate list.

IP Routing Protocol Hierarchies

In the early 1980,s, the Internet was a single network. All routers, which were then called "gateways", shared routing information through the same gateway-to-gateway (GGP) protocol. The routing tables included entries for all IP networks in the Internet.

This configuration caused a number of problems. The routing overhead increased with the number of connected routers. The size of routing table increased with the number of connected networks. The frequency of the routing updates also increased. As the number of routers and links increased the more unstable the network became and hence the more frequent the number of routing updates. As the number of routers increased, so too did the number of different types of routers. Different machines from different manufacturers were increasingly being used. All these machines used their own specific implementation of GGP, which made maintenance and fault isolation almost impossible.

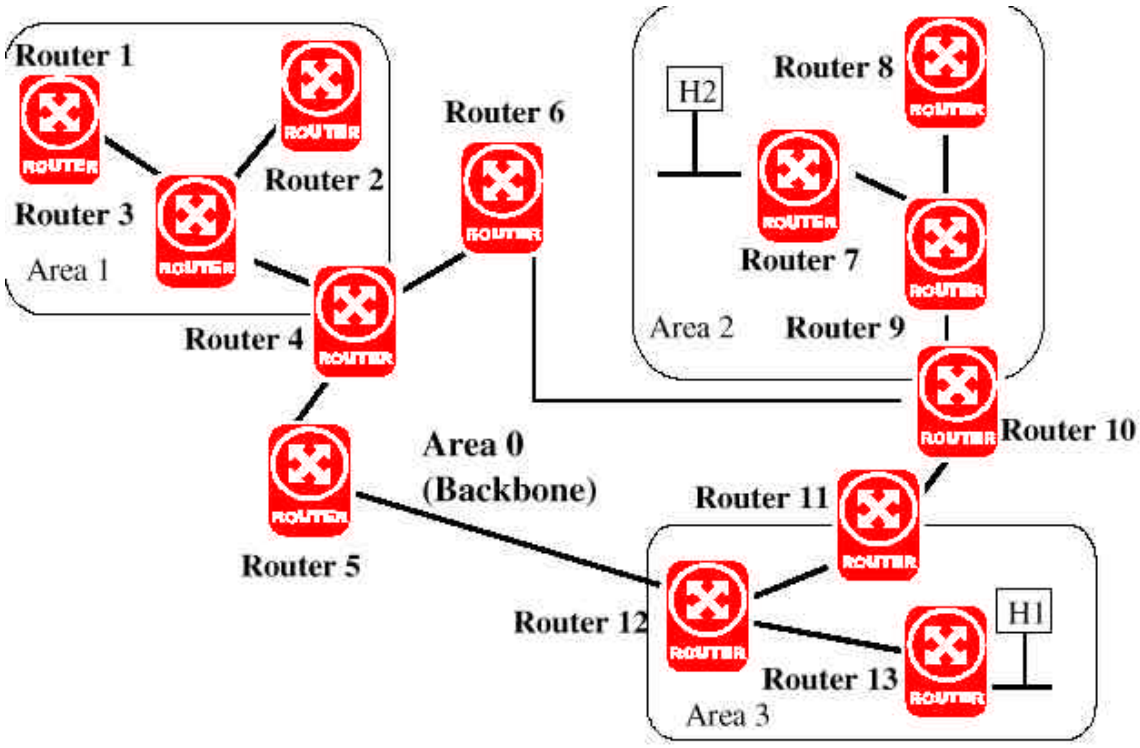
It was decided to split the Internet into a set of Autonomous Systems (AS); each AS comprised of a set of routers and networks under the same administration. One AS was formed from ARPANET and Satnet routers. This formed the core and played a "backbone role". All the other autonomous systems would connect to the backbone using a router called an exterior gateway.

All routers within the same AS are interconnected. These routers exchange routing information. This is normally done by selecting a single routing protocol and running it between all the routers. In 1982 terminology, routers inside an AS were called "interior gateways" and the protocol was an "Interior Gateway Protocol" (IGP). Examples of IGP in use today are RIP, OSPF and IGRP.

These routers can discover information only about the internal networks to which they are directly connected. They must get information about exterior networks through a dialogue with exterior gateways, which are entry points into adjacent autonomous systems. The protocol used between autonomous systems is called Exterior Gateway Protocol (EGP).
EGP organises the exchange of information between two adjacent autonomous systems.

This involves three separate procedures:

- Neighbour acquisition
- Neighbour reachability
- Network reachability



Neighbour acquisition

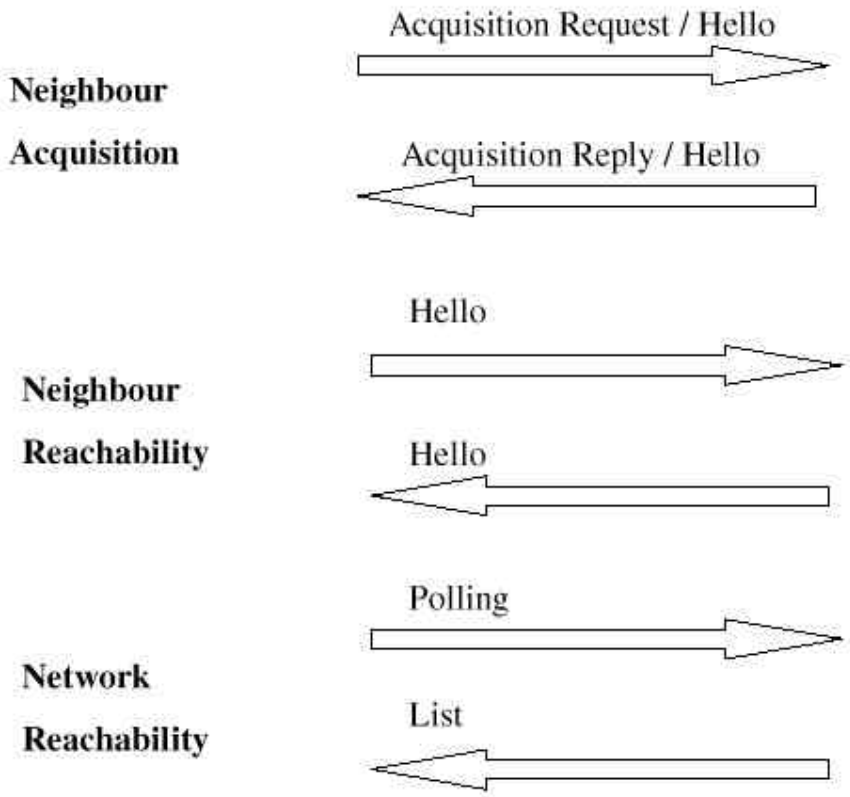
Before exchanging any information and indeed before using any routing information, the adjacent routers must agree to become neighbours for EGP. The neighbour acquisition procedure is a simple "two-way handshake". The router that wishes to become a neighbour sends a "neighbour acquisition request" to its partner, which will reply with an acquisition reply", The partner may also refuse to become a neighbour and reply with a "refusal" message. When a request/reply exchange has been successfully performed, the routers become neighbours.

Neighbour reachability

The purpose of the neighbour reachability procedure is to check that the link to the neighbour is still operational. The router that wants to check reachability sends a "hello" message at regular intervals. The neighbour sends a "I heard you" message in response.

Network Reachability

The purpose of the network reachability procedure is to exchange the list of networks that can be reached through each neighbour. The procedure is based on "polling"- each neighbour, at regular intervals, polls its partner for a list.



RIP

The most widely used interior gateway protocol in today's Internet is Routing Information Protocol (RIP). RIP is a very simple protocol of the distance vector family. The RIP protocol is based on the Bellman-Ford algorithm. RIP allows hosts and gateways to exchange information for computing routes through an IP-based network.

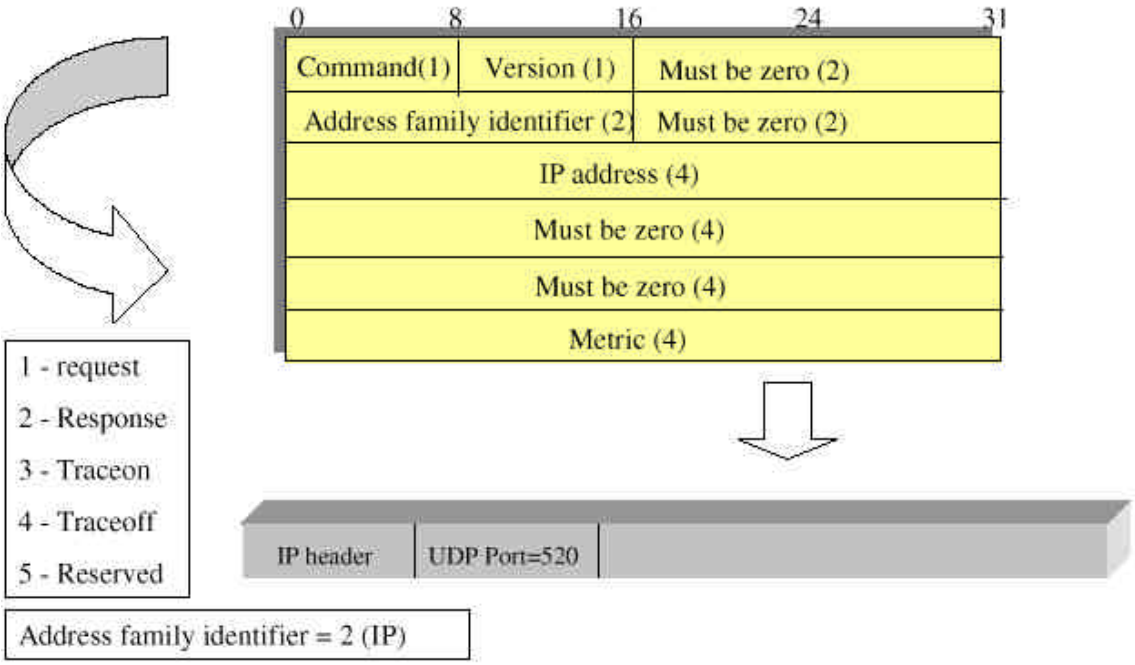
RIP was initially designed as a component of the networking code for the BSD (Berkley System Design) release of UNIX, incorporated into a program called "routed", which is the short for "route management daemon". It is an extremely simple protocol requiring minimal configuration. RIP was built and adopted widely before a formal standard was written. Most implementations were derived from the Berkley code. RIP was documented in RFC-1058 in June 1988 by Charles Hedrick which made it possible for vendors to ensure interoperability.

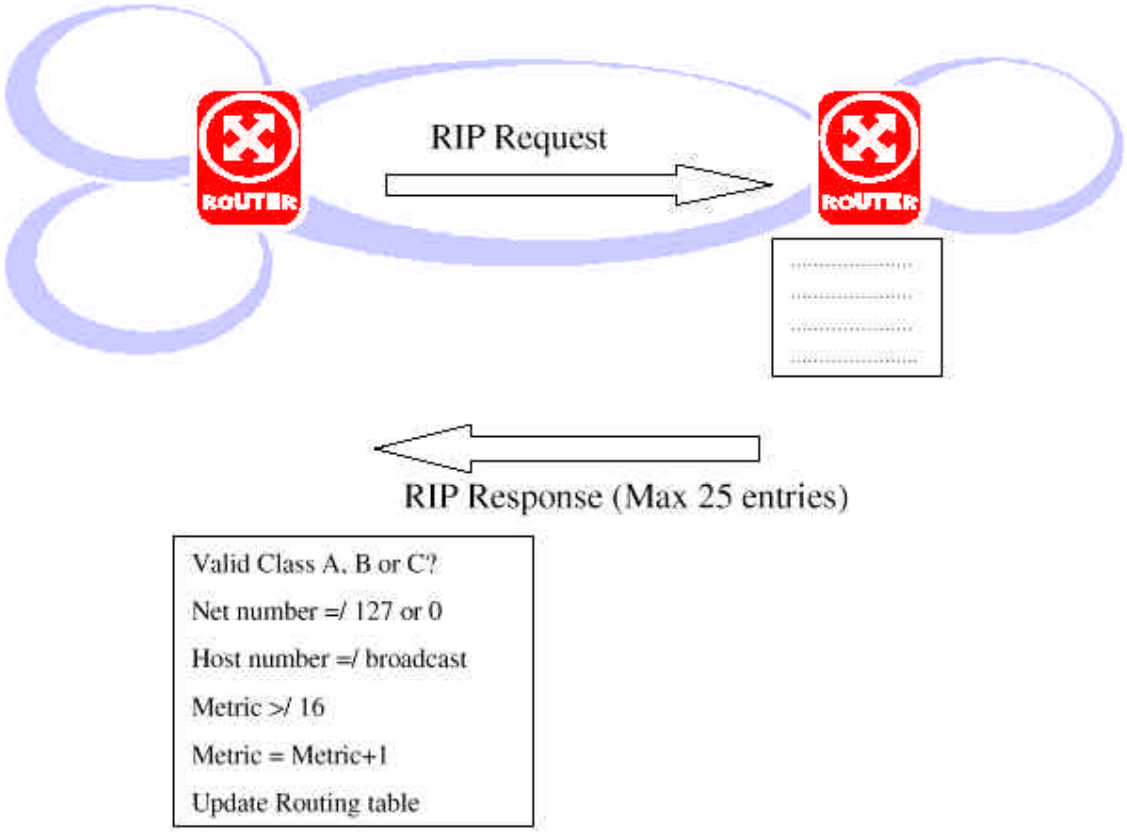
A router running RIP broadcasts a message every 30 seconds. The message contains information taken from the router's current routing database. Each message consists of pairs, each pair contains an IP network address and an integer distance to that network.

The addresses used in RIP are 32-bit Internet addresses. An entry in the routing table can represent a host, a network or a subnet.

By default, RIP uses a very simple metric: the distance is the number of hops to be used to reach the destination. This is normally called the hop count. This distance is expressed as an integer varying between 1 and 15; the value 16 denotes infinity.

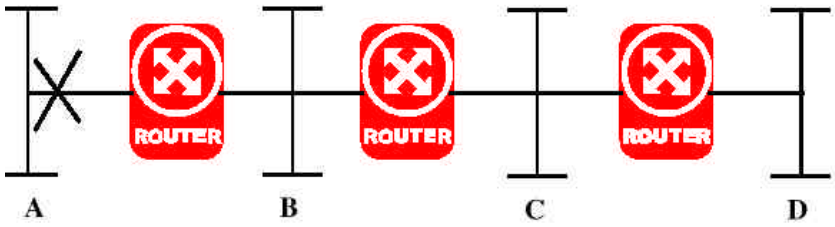
RIP supports both non-broadcast and broadcast networks. RIP packets are carried over User Data Protocol (UDP) and IP. The RIP processes use UDP port number 520 for emission and reception. Packets are normally sent every 30 seconds, or faster in the case of triggered updates. If a route is not refreshed within 180 seconds, the distance is set to infinity and the entry is removed from the table later.





Slow Convergence

The slow convergence problem can make routers wrongly believe they have a connection to a network, after the connection has failed. In the event of a failure, a router stops advertising a route and the protocol must depend on the timeout mechanism before it considers the route unreachable. Once the timeout occurs, the router finds an alternative route and starts propagating that information.



Network	Hops
A	1
B	1
C	2
D	3

Network	Hops
A	2
B	1
C	1
D	2

Network	Hops
A	3
B	2
C	1
D	1

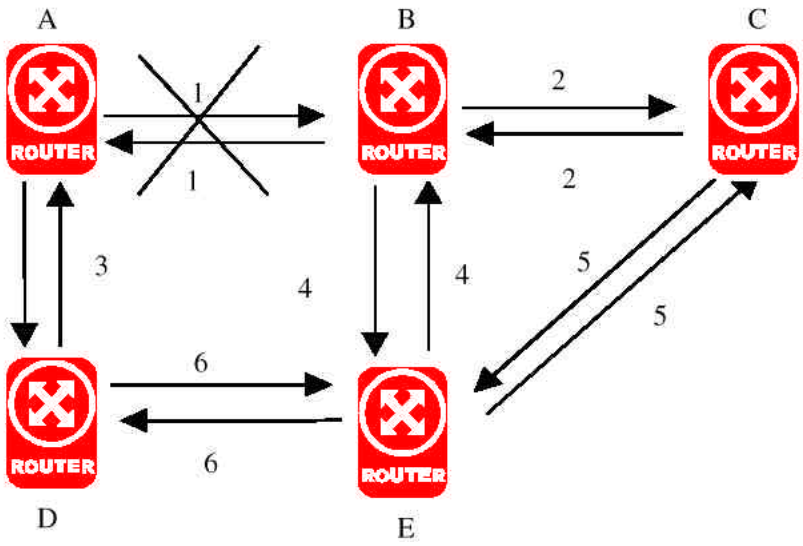
In the example shown in the following figure, Router 1 is aware that its link to network A has failed. However, Router 2 continues to broadcast RIP messages, stating that it can reach network A in 2 hops. Router 1 then assumes it can reach network A via Router 2 and changes its routing table to show the new route. Both Router 1 and Router 2 will continue to exchange RIP messages, increasing the hop count to network A each time, until the count reaches 15 and it is assumed to be unreachable.

Split Horizon

Special techniques have been investigated to try and minimise the effect of loops. Split Horizon is based on a very simple precaution: if Node A is routing packets bound for destination X through node B, it makes no sense for B to try to reach X through A. Split horizon comes in two variations.
In the first variation a router records the interface over which it received a particular route and does not propagate its information about that route back over the same interface.

Split Horizon with Poisonous Reverse

The form known as 'split horizon with poisonous reverse' is more aggressive. The nodes include all destinations in the distance vector message, but set the corresponding distance to infinity if the destination is routed on the link. This immediately kills two-hop loops.



The Use of Timers in RIP

RIP specifies that all routers must timeout routes they learn via RIP. When a router installs a route in its table, it starts a timer for that route. The timer is restarted whenever the router receives another RIP message advertising that route. The route becomes invalid (the metric is set to infinity) if a predefined period of time passes without the route being advertised again.

Every 30 seconds, the output process is instructed to generate a complete response to every neighbouring gateway. When there are many routers on a

single network, there is a tendency for them to synchronize with each other such that they all issue updates at the same time. This can happen whenever the 30 second timer is affected by the processing load on the system. It is undesirable for the update messages to become synchronized, since it can lead to unnecessary collisions on broadcast networks. Thus, implementations are required to take one of two precautions.

- The 30-second updates are triggered by a clock whose rate is not affected by system load or the time required to service the previous update timer.
- The 30-second timer is offset by addition of a small random time each time it is set.

There are two timers associated with each route, a "timeout" and a "garbage-collection time". The garbage collection time is often referred to as the hold down timer. Upon expiration of the timeout, the route is no longer valid.

However, it is retained in the table for a short time, so that neighbours can be notified that the route has been dropped. Upon expiration of the garbage-collection timer, the route is finally removed from the tables.

The timeout is initialized when a route is established, and any time an update message is received for the route. If 180 seconds elapse from the last time the timeout was initialized, the route is considered to have expired and the following takes place.

- The garbage-collection timer is set for 120 seconds.
- The metric for the route is set to 16 (infinity). This causes the route to be removed from service.
- A flag is set noting that this entry has been changed, and the output process is signalled to trigger a response.

Until the garbage-collection timer expires, the route is included in all updates sent by this host, with a metric of 16 (infinity). When the garbage-collection timer expires, the route is deleted from the tables.

OSPF

The development of the Open Shortest Path First (OSPF) routing protocol began in 1987. To understand the goals of the OSPF working group, one needs to consider the nature of the Internet of 1987. It was largely an academic and research network, funded by the US government. Much of the Internet used static routing; Autonomous Systems employing dynamic routing used (Routing Information Protocol) RIP, while External Gateway Protocol (EGP) was used between Autonomous Systems. OSPF version 2 is documented in RFC 2328.

Both were experiencing problems. As the size of Autonomous Systems grew and the size of the Internet routing tables increased, the amount of network bandwidth consumed by RIP updates was increasing, and route convergence times were becoming unacceptable as the number of routing changes also increased.

Other initial functional requirements for the OSPF protocol included the following:

- A more descriptive routing metric. A configurable link metric whose value ranges between 1 and 65,535 was chosen. This removed network diameter limitations and allowed factors such as bandwidth, cost and delay to be used when configuring routing systems.
- Equal-cost multipath. OSPF can discover multiple best paths to a given destination. With equal cost multipath, a router potentially has several available next hops towards any given destination.
- Routing hierarchy. This enables us to build very large routing domains, on the order of many thousands of routers.
- Support for more flexible subnetting schemes. OSPF supports variable length subnet masks (VLSMs), whereby a class A, B or C address can be divided into unequal subnets.
- Security. OSFP packets have a space reserved for authentication. By authenticating received OSPF packets, a router would have to be given the correct key before it could join the OSPF routing domain.

Link State Protocol

The Open Shortest Path First (OSPF) routing protocol is a link state algorithm, that adjusts to network changes more quickly than RIP and is more robust.

Each router updates the rest of the network with information on the direct connections it has to its neighbours.

OSPF routers advertise their routing information in Link-State Advertisements, or LSAs. OSPF routers broadcast LSAs to their neighbours only when a network change has occurred. They also send an update following a large interval, typically every hour. LSAs are described in detail later in this chapter.

The Backbone of the Autonomous System

When an OSPF routing domain is split into areas, all areas are required to attach directly to a special area called the OSPF backbone area. The backbone area always has the Area ID 0.0.0.0. The OSPF backbone always contains all

Area Border Routers. The backbone is responsible for distributing routing information between non-backbone areas. ABRs run multiple copies of the basic algorithm, one copy for each attached area. ABRs condense the

topological information of their attached areas for distribution to the backbone. The backbone in turn distributes to other areas.
The backbone must be contiguous. However it need not be physically contiguous. Backbone connectivity can be established and maintained through the configuration of virtual links.

Each ABR summarises the topology of its attached non-backbone areas for transmission on the backbone and hence to all other ABRs. An ABR then has the complete topological information concerning the backbone and the area summaries from each of the other ABRs. From this information, the router calculates paths to all inter-area destinations. The router then advertises these paths into the attached areas. This enables the area's routers to pick the best exit router when forwarding traffic to inter-area destinations.

Inter-area Routing

When routing a packet between two non-backbone areas the backbone is used. The path that the packet travels is broken into three contiguous pieces; an intra-area path from the source to the ABR, a backbone path between the source and destination areas, and then another intra-area path to the destination. Looking at this in another way, inter-area routing can be pictured as forcing a star configuration on an Autonomous System, with the backbone as hub and each of the non-backbone areas as the spokes.

AS Boundary Routers (ASBR)

When an OSPF routing domain is connected to an external network, a special router known as the Autonomous System Boundary Router (ASBR) is used to interconnect between the OSPF routing domain and the external routing domain. Information is distributed verbatim to every participating router. The paths to each ASBR are known by every router in the AS. ASBRs may be internal or ABRs and may or may not participate in the backbone. The ASBR leaks OSPF summary-LSAs from the external routing domain to the OSPF routing domain and visa versa.

OSPF Area Types

OSPF supports three area types. These are:

- Transit Areas
- Stub Areas
- Not So Stubby Areas (NSSA)

Transit Area

A transit area includes any area capable of propagating or originating AS external LSAs. (AS external LSAs are OSPF external-LSAs (Type 5) that are originated by AS Border routers and propagated throughout the OSPF routing domain). The backbone area is always, by definition, a transit area.

Stub Area

In some Autonomous Systems, the majority of the link state database may consist of AS-external-LSAs. An OSPF AS-external-LSA is usually flooded throughout the entire AS. However, OSPF allows certain areas to be configured as stub areas.

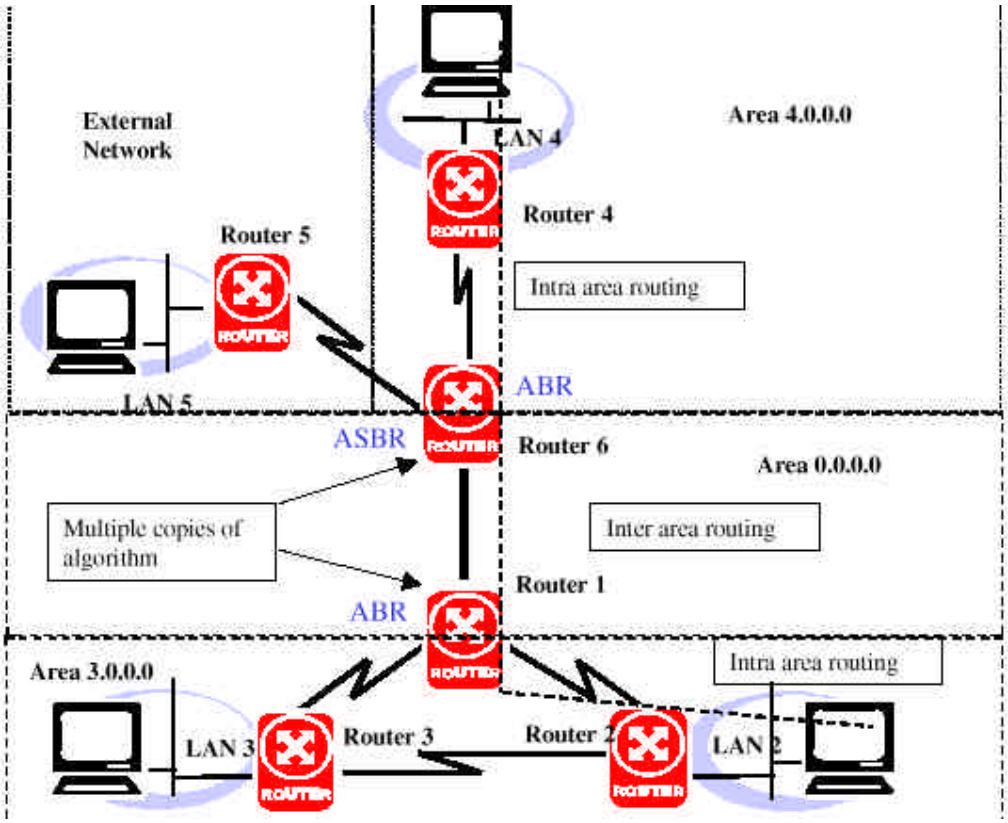
Stub areas do not import external route information. OSPF stub areas cannot

contain ASBRs. External LSAs (Type 5) are not propagated into the area nor may a stub area originate External LSAs. Instead, network traffic to destinations not local to the area or AS is directed to the closest area border router advertising a default route. There is a default route per area. This reduces the link-state database size and therefore the memory requirements, for a stub area's internal routers.

Not So Stubby Areas

The NSSA (Not-So-Stubby-Area) defines a new OSPF area similar to the stub area in that external LSAs (Type 5) are not propagated into the area nor may they originate in a stub area (via an ASBR). However, the area may contain an AS border router that may inject NSSA LSAs (Type 7) into the area. These NSSA LSAs (Type 7), which are virtually the same syntax as an external-LSA (Type 5), may then be propagated into other areas by the ABRs as regular external-LSAs (Type 5).

Several configuration options must be completed for an area to be configured as a NSSA. First, the area must be configured as NSSA on all internal routers or routers that have an interface in the area. Any ASBR within the area must be configured to import externally derived routing information (such as RIP, BGP, static routing, and so on). This is done by adding import policies and configuring the router as an ASBR router. If NSSA LSAs (Type 7) are to be propagated into the backbone area, the ABR routers must contain area summarisation policies configured to propagate NSSA LSAs (Type 7) as external LSAs (Type 5) outside of the area. Summarisation policies for networks contained within the area (for network summary-LSAs) are still required for area summarisation.



The Protocols within OSPF

OSPF is composed of three subprotocols. These are:

- The Hello protocol
- The Exchange protocol
- The Flooding protocol

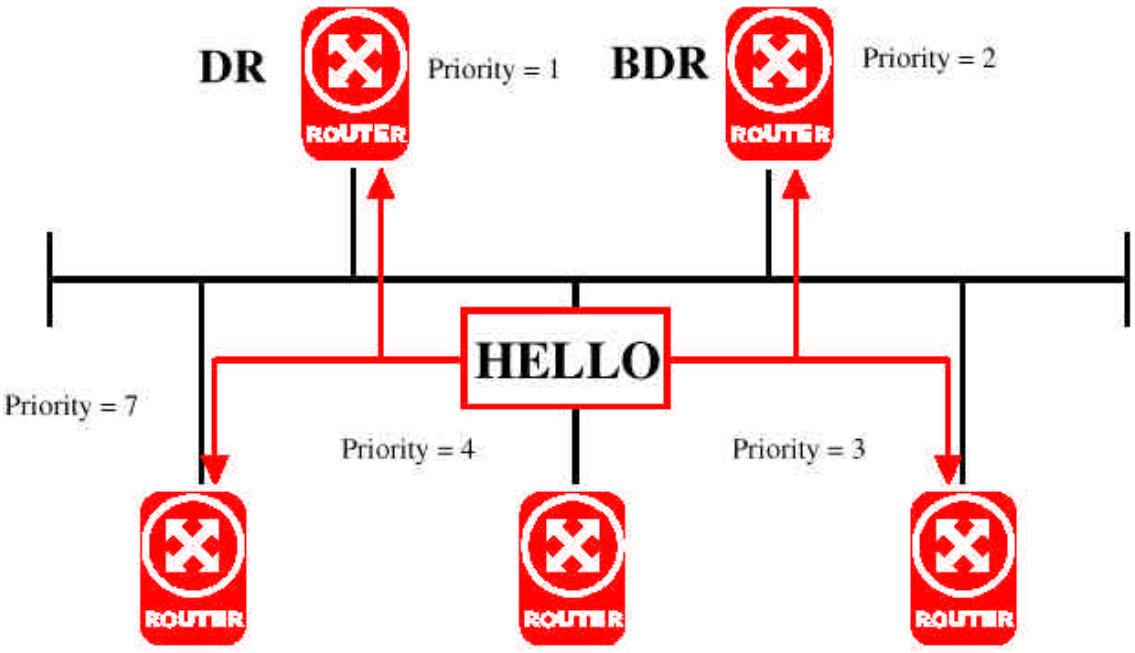
The Hello Protocol

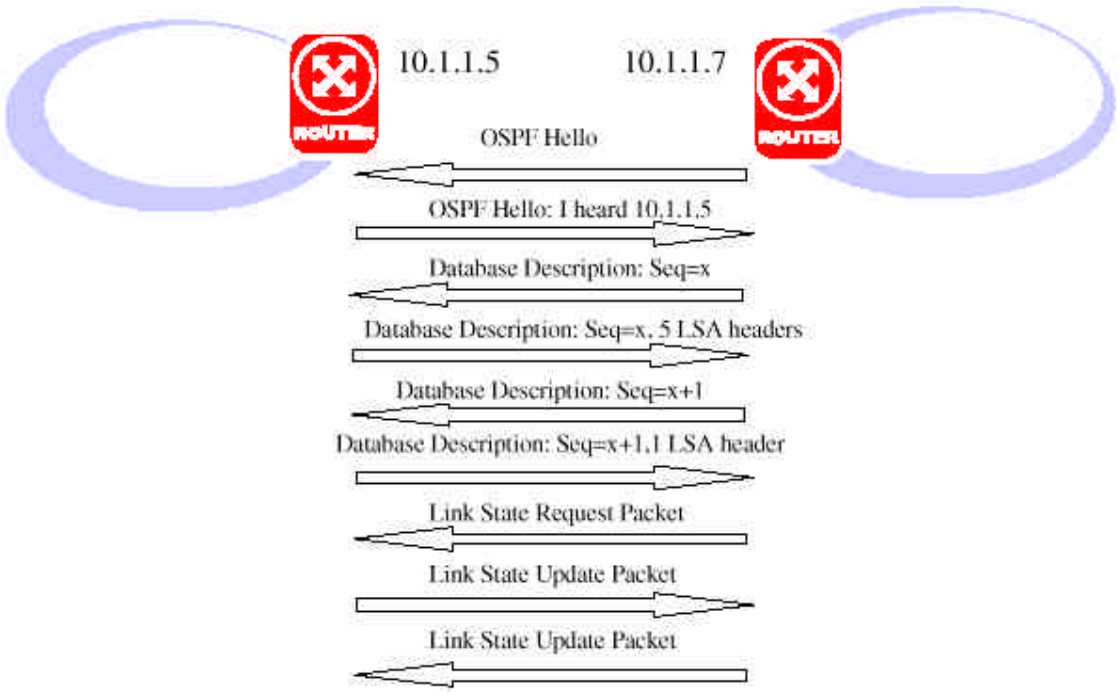
OSPF Hello packets have an OSPF header with the type field set to 1. The Hello protocol is used for two purposes:

- To check that the links are operational
- To elect the Designated Router (DR) and the Backup Designated Router (BDR)

A router discovers its neighbours by periodically sending OSPF Hello packets out all its interfaces. By default, a router sends out these packets at 10 second intervals. This interval can be configured by the network administrator. A router learns about its neighbour when it receives a Hello packet from its neighbour. If no Hello packet is received within a certain time interval (40 seconds by default) the router stops advertising the connection to the router and starts routing data packets around the point of failure.

The OSPF Hello protocol also establishes that the neighbouring routers are consistent in the following ways. The Hello protocol ensures that the link is bidirectional.





The Exchange Protocol

When two routers have established two-way connectivity on a point-to-point link, they must synchronise their databases. The initial synchronisation is performed by the exchange protocol. The flooding protocol is then used to maintain the two databases in synchronisation. The routers must ensure that their link-state databases are synchronised before forwarding traffic over the connection.

The OSPF exchange protocol is asymmetric. The first step of the protocol is to select a “master” and a “slave”. After agreeing on these roles, the two routers will exchange the description of their databases, and each lists the records that will be requested at a later stage. The exchange protocol uses database description packets.

Database Synchronisation

Instead of sending the entire database to the neighbour when the connection comes up, an OSPF router sends only its LSA headers, and the neighbours request the most recent LSAs. This is called database exchange. This procedure is more efficient than sending the entire database. The link-state headers are sent in a series of OSPF database description packets. Only one database description packet can be outstanding at any one time; the router sends the next database description packet only when the previous one is acknowledged through reception of a properly sequenced database description packet from the neighbour.

When the entire sequence of database description packets have been received, the router knows the link-state headers of all the LSAs in its neighbours link state database. The router also knows which of its neighbour's LSAs it does not have and which of it's neighbour's LSAs are more recent. The router then sends link state request packets to the neighbour requesting the desired LSAs, and the neighbour responds by flooding the desired LSAs in link

state update packets.

Once this process is complete, the routers declare the connection synchronised and advertise it for use by data traffic. At this point the neighbour is said to be fully adjacent to the router.

The router may have Max Age LSAs in its link-state database as database exchange begins. Since Max Age LSAs are in the process of being deleted from the database, the router does not send them in Database Description packets to the neighbour.

OSPF Link Status Request Message Format

After exchanging database description messages with a neighbour, a router may discover that parts of its database are out of date. To request that the neighbour supplies updated information, the router sends a link status request message. The message lists specific links. The neighbour responds with the most up-to-date information it has about those links. The three fields, Link Type, Link Id and Advertising Router are repeated for each link. More than one request message is sent when the list of requests is long.
Link Status request packets are OSPF packets with the type field set to 3. A router that sends a Link State request packet knows exactly the precise instance of the database it is requesting.
Each LSA requested is specified by its LS type, Links State ID, and Advertising Router.

The Flooding Protocol

The flooding protocol is used to maintain the two databases in adjacent routers in synchronisation.
When a link changes state, the router responsible for that link will issue a new version of the link state. This is carried in link state update messages.

OSPF Link Status Update Message Format

Each link status update message has a special header. Link State Update packets are OSPF packets with the type field set to 4. The OSPF header is followed by an indication of the number of advertisements and by the link state advertisements themselves. The values used in the header are the same as in the database description message.

The body of the Link State Update packet consists of a list of LSAs. Each LSA begins with a common 20 byte header. LSA packets are explicitly acknowledged. This is achieved through the sending and receiving of Link State Acknowledgement packets.