**Routers**

A router is an [operates at the network layer of the OSI reference model. Routers are used to connect two or more](https://erg.abdn.ac.uk/users/gorry/course/intro-pages/es-is.html) [IP networks](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip.html).

A router consists of a computer with at least two [network interface cards](https://erg.abdn.ac.uk/users/gorry/course/lan-pages/nic.html) supporting the [IP protocol](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip.html). The router receives packets from each interface via a [network interface](https://erg.abdn.ac.uk/users/gorry/course/lan-pages/nic.html) and forwards the received packets to an appropriate output network interface. The key points here are that routers in the network have a perspective of the correct path to use to reach a destination.

A router receives packets from each interface via a network interface. Received packets have all link protocol headers removed.

The router uses the information in the [IP header](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip-packet.html) to decide whether and where to forward each received packet, and which network interface to use to send the packet. Most packets are forwarded based on the packet's [IP destination address](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip-address.html), along with routing information held within the router in a routing table.

A router does not by default forward a packet with an IP broadcast address. Protocols using broadcast packets normally only operate within a single broadcast domain. (Routers can sometimes be specifically configured to support this.)

Once the transmit interface is selected, the router checks the Maximum Transfer Unit (MTU) of the interface. If the packet size is less than or equal to the MTU, the processing continues. All IP packets contain a TTL value that determines the number of router hops that a packet may be routed, this is decremented by most routers (although routers can be configured to reduce the value by more than 1) It is used to prevent routing loops, and ensures topologies with loops do not result in packets that circulate indefinitely.

At the output interface, the packet (together with a new link layer header) is placed into a transmit queue until the link layer processor is ready to transmit the packet. This, like the receive queue. Each out-going packet requires a new link layer protocol header to be added (encapsulation) with the destination address set to the next system to the receive the packet. The link protocol controller also maintains the hardware address table associated with the interface. This usually involves using the Address ReResit Exam Solution Protocol (arp) to find out the hardware (MAC) addresses of other computers or routers directly connected to the same cable (or LAN). The packet is finally sent using the media interface with the hardware address set to the next hop system. In this way, transmitted packets have a new link protocol header added prior to transmission of each packet.

IPv4 Packets larger than the interface MTU cannot be sent. They can be fragmented by the router into two or more smaller packets. If an IPv4 packet is received which has the Don't Fragment (DF) bit set in the [packet header](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip-packet.html) or uses IPv6, the packet is not fragmented, but is instead discarded. In this case, an [ICMP error message](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/icmp.html) is returned to the sender (i.e. to the original packet's IP source address) informing it of the interface's MTU size. This forms the basis for [Path MTU discovery (PMTU)](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/mtu.html).

**Routing and Filter Tables**

The routing and filter tables resemble similar tables in link layer bridges and switches. Except, that instead of specifying link hardware addresses ([MAC addresses](https://erg.abdn.ac.uk/users/gorry/course/lan-pages/mac.html)), the router table sepcify network ([IP addresses](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip-address.html)). The routing table lists known IP destination addresses with the appropraite network interface to be used to reach that destiantion. A default entry may be specified to be used for all addresses not explicitly defined in the table (this is very common in routers close to the edge of the networ, where the default routes packets towards the Internet backbone).

A filter table may also be used to ensure that unwanted packets are discarded. The filter may be used to deny access to particular protocols or to prevent unauthorised access from remote computers by discarding packets to a specified destination address. Routers at the edge of ISP networks also often perform filtering of the IP source address, as a way to prevent "spoofing" of addresses belonging to other networks.

A router forwards packets from one IP network to another IP network. Like other systems, it routes based on the longest-prefxi match of the IP addresss in the routing table.

One exception to this rule is when a router receives an IP packet to a network broadcast address. In this case, the router will process the packet internally (to see if it needs to respond) and then discards the packet. Forwarding a broadcast packet can lead to severe storms of packets, and if uncontrolled could lead to network overload.

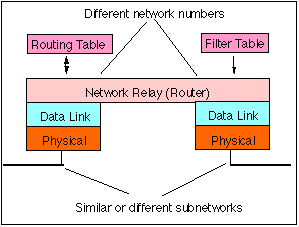
At the output interface, the packet (together with a new link layer header) is placed into a transmit queue until the link layer processor is ready to transmit the packet. This, like the receive queue. Each out-going packet requires a new link layer protocol header to be added (encapsulation) with the destination address set to the next system to the receive the packet. The link protocol controller also maintains the hardware address table associated with the interface. This usually involves using the Address ReResit Exam Solution Protocol (arp) to find out the hardware (MAC) addresses of other computers or routers directly connected to the same cable (or LAN). The packet is finally sent using the media interface with the hardware address set to the next hop system.

**Latency**

A router introduces delay (latency) as it processes the packets it receives. The total delay observed is the sum of many components including:

* Time taken to process the frame by the data link protocol
* Time taken to select the correct output link (i.e. filtering and routing)
* Queuing delay at the output link (when the link is busy)
* Other activities which consume processor resources (computing routing tables, network management, generation of logging information)

The router queue of packets waiting to be sent also introduces a potential cause of packet loss. Since the router has a finite amount of buffer memory to hold the queue, a router which receives packets at too high a rate may experience a full queue. In this case, the router ahs no other option than to simply discard excess packets. If required, these may later be retransmitted by a transport protocol.

**

*Architecture of a router*

Routers are often used to connect together networks which use different types of links. The optimum (and maximum) packet lengths (i.e. the [maximum transmission unit (MTU)](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/mtu.html)) is different for different types of network. A router may therefore uses IP to provide segmentation of packets into a suitable size for transmission on a network.

IP routers allow alternate paths to be dynamically selected when routers forward packets. Adding parallel paths through the network adds robustness to failure - if one link, interface (or router) fails, an alternate path may be selected restoring end-to-end connectivity. All IP packets contain a TTL value that determines the number of router hops that a packet may be routed. A router decrements the hop count (IPv4 TTL or IPv6 HopCount) field each time it forwards a packet. If a packet gets stuck in a loop the TTL counts down to zero - at which point the packet is automatically discarded.

Associated protocols perform network error reporting ([ICMP](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/icmp.html)), communication between routers (to determine appropriate routes to each destination) and remote monitoring of the router operation (network management).

The [operation](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/router-opn.html) of a simple modern router is described on a separate page. If you want to know how the router actually works click [HERE](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/router-opn.html).

# Operation of a Router:

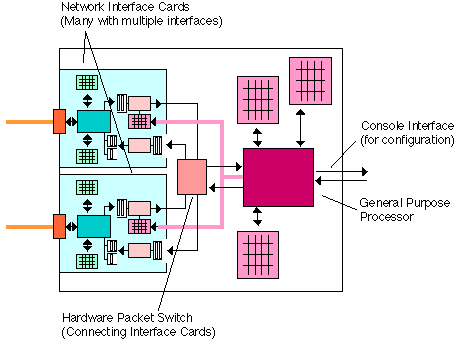
# Operation of a Router

The (current) Internet infrastructure consistss of an interconnected set of networks (known as autonomous systems (AS)) that are independently administrated. Packets are forwarded between links within an AS and between different AS's using router. There are somewhere around 10,000 different AS currently in use. A modern [router](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/router.html) is a complex piece of equipment. The outside of the equipment is usually very simple, consisting of a number of network interface ports (shown on the left in orange in the figure below) to which cables may be connected and a few indicator lights to indicate that the router is functional.

Routers can be divided into three key components: a routing engine, a forwarding engine and a management agent. The function of the routing engine is to process routing information (exchanged between routers using a routing protocols such as the Border Gateway Protocol, BGP) so as to compute routes (using a shortest path algorithms) that are stored in routing information bases (RIB) and that are composed by a destination, a next-hop interface, and a metric. Routing entries are subsequently used to populate the forwarding information base (FIB) whose entries are used by the forwarding engine. The function of the forwarding engine is to transfer incoming traffic to an outgoing interface directed towards a router closer to the traffic destination by performing a longest match prefix lookup using the incoming traffic destination address. This forwarding process is connectionless implying that at each hop the forwarding decision is taken independently for each datagram.

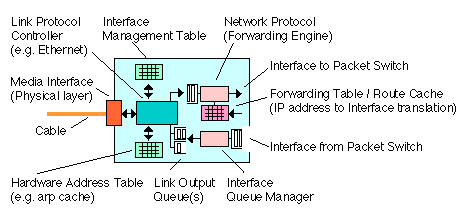
**Overview**

Most routers also have a serial connector to which a terminal (or a modem) may be connected, known as the "**Console Port**" (shown to the right in the figure below). This port is usually used to control the router configuration when the router is first installed. It may be the only port which is allowed to configure the filter table (used to prevent unauthorised access between the connected networks).



*Block diagram of a complete router showing the interfaces Interfaces to the network (over which packets are received or transmitted) are shown left, and the control interface (used to set up and reconfigure the router) is shown right.*

In the simplest case, the processing of packets is implemented in the general purpose processor which implements all the algorithms. More advanced routers may separate "**forwarding**" (the tasks of moving packets from one interface to another) from "**routing**" (the task of determining the best path through the network) and include a number of processors capable of performing these tasks. A router interface card resembles the [LAN Network Interface Cards (NICs)](https://erg.abdn.ac.uk/users/gorry/course/lan-pages/nic.html) used in PCs except that the card is normally of a higher specification (faster packet processing). The very first routers were designed used standard network interface cards, but modern high performance routers use special high performance interface cards and may also include a "Forwarding Engine" on-board the card which speeds the operation.



*Router Network Interface Card*

Received packets are processed by the **link layer protocol controller**, which handles the link layer protocol (e.g. [HDLC](https://erg.abdn.ac.uk/users/gorry/course/dl-pages/hdlc.html), [Ethernet](https://erg.abdn.ac.uk/users/gorry/course/lan-pages/enet.html)) used over the physical link (cable). This also checks the received frame integrity (size, checksum, address, etc). Valid frames are converted to packets by removing the link layer header and are queued in the receive queue. This is usually a First-In-First-Out (FIFO) queue, often in the form of a ring of memory buffers.

The buffers are passed (drained) into the input to the **forwarding engine**. This takes each buffer, one at a time, and removes it from the interface receiver. The packet is then forwarded to an appropriate output interface, corresponding to the "best" path to the destination specified in the destination address of the IP packet header.

At the output interface, the packet (together with a new link layer header) is placed into a transmit queue until the **link layer processor** is ready to transmit the packet.This, like the receive queue, is a FIFO queue, and usually also takes the form of a ring of memory buffers.

Each out-going packet requires a new link layer protocol header to be added ([encapsulation](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip-enet.html)) with the destination address set to the next system to the receive the packet. The link protocol controller also maintains the **hardware address table** associated with the interface. This usually involves using the [Address Resolution Protocol (arp)](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html) to find out the hardware ([Medium Access Control](https://erg.abdn.ac.uk/users/gorry/course/lan-pages/mac.html)) addresses of other computers or routers directly connected to the same cable (or [LAN](https://erg.abdn.ac.uk/users/gorry/course/intro-pages/lan.html)). The packet is finally sent using the media interface with the hardware address set to the next hop system. When complete, the buffer (memory) allocated to the frame, is "freed", that is, it is returned as an empty buffer to the receive queue, where it may be used to store a new received packet.

You may think from this that the job of forwarding is not too difficult, and involves a lot of copying of the packet data from one place to another. You would be wrong on both counts! Forwarding actually involves lots of decisions. Modern routers avoid copying the data in a packet if at all possible - this is a significant processing cost, and may easily slow down a router to a very low throughput. Instead, where ever possible, the router will leave the packet data in the same place and instead pass information about ***where*** a packet is stored in memory.

Introduction to Routing Protocols

Dynamic routing protocols may be divided into interior (intradomain) and exterior (interdomain) protocols.Exterior

Two types of dynamic routing protocol are commonly used:

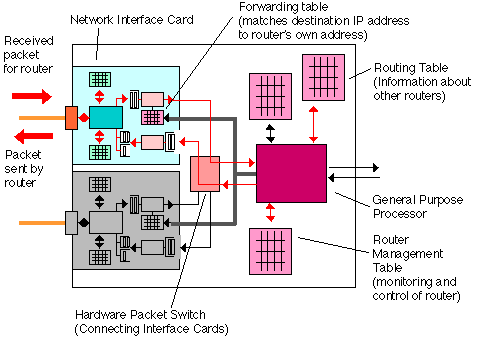
* Interior Routing Protocol: The most common exterior routing protocol is the Border Gateway Protocol (BGP) [RFC4271] and [RFC2545]. BGP is often used to support multi-homing in operator networks to provide path redundancy. It is also used to support inter-domain multicast routing between PIM-SM Domains. Interior protocols may be subdivided into Distance Vector (e.g., RIP [RFC2453] for IPv4 or RIP-NG [RFC2080] & [RFC2080] for IPv6) and Link-State (e.g., OSPF for IPv4 [RFC2328] and for IPv6 [RFC5340]). RIP is typically used only in small networks where no policy control is required and where overhead is not a major concern. OSPF requires more processing power than RIP but exchanges more compact data.
* Exterior Routing Protocol: The most common exterior routing protocol is the Border Gateway Protocol (BGP) [RFC4271] and [RFC2545]. BGP is often used to support multi-homing in operator networks to provide path redundancy. BGP-4 distributes routing reachability information between autonomous systems and sends only incremental information. It uses TCP connections and sends keep-alive packets every 60 seconds. A routing protocol may be used to import routing information concerning the networks that are connected. Many network operators implement policies to control the routes that are imported to protect the routing information in the RIB, although this is not recommended for link-state protocols, such as OSPF.

Although all routers provide the above functions, there are significant differences in their feature sets, often determined by where they are placed within the network:

* Enterprise routers often have expanded feature sets (usually use OSPF and may support VPNs, VLANs, access control, network management, VoIP services, firewall functions, L2 virtualisation, etc).
* Provider Edge (PE) routers often used BGP and are usually configured to enforce policies (e.g., QoS, access control). They may function as a LER for MPLS, terminating VPN services, etc. They are managed devices that usually support redundant operation.
* Provider routers are simpler, and usually use BGP or MPLS LSR to control forwarding, and usually support advanced management and redundant operation.
* Core routers are usually highly optimised for forwarding speed, high-speed optical interfaces and redundant operation.

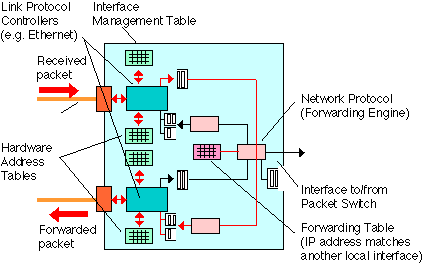
**Introduction to Forwarding**

This section gives a simple description of the forwarding process. After determining the link layer frame is valid, the forwarding engine then starts processing the network layer information. It reads the network layer ([IP](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip.html)) packet headers and [checks](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip-processing.html) various parts of the [header](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip-packet.html), to ensure the packet is not damaged or illegal. It then uses a local **Forwarding Table** (known as the "**Forwarding Information Base (FIB)**") to identify where in the network the packet should be routed to (i.e. which output interface should be used).



*Forwarding a received packet to an output interface via the packet switch.*

Once the appropriate output interface has been identified, the forwarding engine then requests the **packet switch** to form a connection to the appropriate output interface. The packet is then moved through the router to the output network interface controller. Although large routers actually implement a switch as a hardware component, most smaller routers do not actually contain a "real" switch. In other routers, the switch takes the form of a shared memory data structure in which all received packets are stored. The switching operation therefore consists of removing a pointer from the receive queue, and copying the value of the pointer to the appropriate transmit queue. In some cases, the entire packet data is copied from one bank of receive memory to another transmit memory using a computer bus.



*Packets may be directly forwarded from one controller to the other (sometimes known as "Fast Switching"). This occurs when the forwarding table has a match for the IP destination address of the received packet which indicates the packet should be sent out using one of the other interfaces on the same card, and the packet does not require any special IP processing. This type of forwarding is very efficient, since it causes very little load on the router processor.*

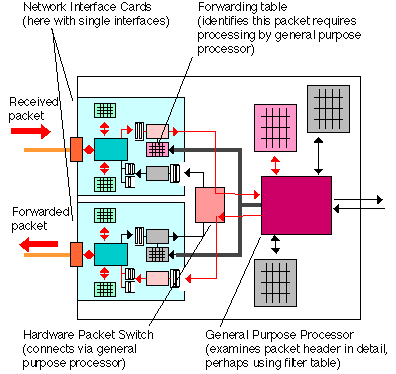
The operation of the router is controlled by one or more **general purpose processor** which is usually similar to to a standard high-end PC CPU. The processor's performs various tasks, which may be divided into three groups:

* Process Switching
* Fast Switching
* Routing

The first first two tasks are considered first, and concern packet forwarding. The final task, may in some routers be performed by a separate processor.

**Process Switching**

Every router allows packets to be handled by a CPU using software which implements the various protocols which define the IP network layer. This processing is known as the "Slow Path", (it is typically much slower than processing by Fast Switching, described next). Although slower, the general purpose CPU is however able to perform more sophisticated processing (e.g. [packet fragmentation](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/mtu.html)), and is therefore more flexible. It is also software-based, and therefore can easily be updated as new features are required. In practice, only occasional packets travel on the slow path using process switching, but since it is universally implemented, it will be described first.



*Slow path processing of a packet routed via the processor. This type of operation is required to handle more complex processing or to implement firewall filtering to allow an administrator to control access to and from a LAN to a WAN.*

The most complicated way to process a packet received by a router is Process Switching. In this scheme, the following (roughly speaking) series of tasks are performed:

Receive Processing by Interface Card

Received packet is checked  
The packet is placed in a shared memory pool.

An interrupt called on the Switching Processor CPU.

Interrupt Handling by Switching Processor (Process Switched)

The CPU records the packet's address in memory and schedules a process to find out where the packet is to be sent to (forwarding).  
The interrupt returns.

Switching by Switching Processor (Process Switched)

Some time shortly later (after all interrupts have completed, and CPU has completed its current set of tasks) the switching processor looks at the packet  
The CPU checks whether the packet contains any options - if it does these are processed.  
IP destination address is extracted.  
If the packet is for this router (IP destination address matches a router interface address), it is forwarded to the appropriate task (TCP,UDP,etc).

If it is for a remote computer, the current forwarding table is searched to find the IP address of the next hop router.  
This may require a number of route lookups in some cases.  
The corresponding output interface is found.  
The link layer address of the next hop router is now found (e.g. a MAC address found by looking in the arp cache).  
If there is no entry, an arp request may be sent (e.g. for an Ethernet interface) and the packet remains queued until a response (or timeout) is received.  
The switching processor now knows where to send the packet and the new link layer header to use. It must also check the size of the packet to see if the MTU of the output interface is large enough, if not, the packet has to be fragmented.  
The switching processor caches these values in the forwarding cache (see fast switching later).  
The link layer address is added and the packet is linked into the list of frame to be sent on the appropriate interface.  
The switching processor informs the corresponding interface processor that the packet is waiting.

Transmit Processing by Interface

When the interface transmit process is next idle (i.e. when all the frames ahead of this frame in the transmit interface queue have been sent), the interface processor transmits the frame.  
The interface processor interrupts the process switching CPU to tell it the frame has been sent.

Interrupt Handling by Switching Processor (Process Switched)

The memory buffer is now freed by returning it to the set of buffers available for new packets received.

The switching processor also updates the statistics count held in the router management information base.

The above algorithm is somewhat simplified. In reality, routers also need to be handle tunnels (packets carrying other packets inside them), quality of service features (such as priority for some packets) and various techniques to mitigate the effects of overload.

**Fast Switching**

In many cases, a number of packets are sent by the same end system to the same destination IP address. Using process switching, each of these packets is handled independently - just as one would imagine for a connection-less protocol. But, this processing is costly when performed in this way. In fact, once one packet has been process switched, the router now understands the way to switch all successive packets to the same destination. That is the reason why, the process switching cached (or stored a copy of the outcome) the forwarding decision after it has been made.

Using the cached information (IP destination address, port number, link address, and any other necessary details), can significantly speed-up the forwarding by by-passing many decisions. This is known as the "Fast Path" or "Fast Packet Forwarding" and is outlined below:

Receive Processing by Interface Card

This is as in process switching.

Interrupt Handling by Switching Processor (Fast Switched)

The Switching Processor checks whether the packet contains any options - if it does it uses process switching (i.e. a task is scheduled to process the packet).

The Switching Processor checks whether the IP destination address is in the forwarding table - if it is not, it uses process switching.

The Switching Processor checks the forwarding cache to see if there is an entry for the destination IP address, if not, it uses process switching.

If fragmention is required (or anything apart from simple forwarding), it uses process switching.  
The Switching Processor now knows where to send the packet (interface) and the new link layer header to use by taking values from the forwarding cache.  
The link layer address is added and the complete frame is linked into the list of frame to be sent by the appropriate interface processor. If the process-switched output queue is empty *and* there is space in the transmit interface FIFO ring, the frame may be placed directly in the transmit interface output queue.

The Switching Processor informs the interface processor that the packet is waiting.  
The interrupt returns.

Transmit Processing by Interface Card

As in process switching.

Interrupt Handling by Switching Processor (Process Switched)

As in process switching.

This scheme is much faster than process switching. However, the fast path may only be used for packets which have previously been sent to the same address. The first packet is therefore always process switched.

In practice, it is unwise to keep any cached information for too long. This prevents the information becoming stale (e.g. when a router fails). CISCO (a well-known router supplier) recommends a small part of the cache (e.g. 1/20th) is deleted every minute. Since it is very computationaly expensive to find all the entries in the table referring to a single route of link layer address, the entire table is deleted (purged) whenever the routing table or an interface arp table changes.

Fast switching is very effective at the edge of networks, or within private networks (where there are comparatively few destination addresses and routes). As the number of entries in the forwarding information base increases, the impact of purging the table becomes more and more significant. The rate of purges increases with the number of routers being communicated with. To help this, as the size of the forwarding table increases, the proportion of addresses may need to be deleted (e.g. 1/5th for a FIB>200KB). Fast switching provides little advantage at the centre of the internet (core).

N.B.1)The output queue is always used if there are any packets waiting there, this helps reduce the re-ordering of packets when packets for the same destination are being both process and fast switched.

N.B. 2) Some interfaces use a number of queues (from a few to several thousands). Normally one queue is reserved for network control data (such as routing packets) to ensure these are never delayed (in overload, such packets are particularly important since failure to receive them can impact the stability of the network). If a number of additional queues are being used by the interface, the packet is placed in the output queue, rather than the FIFO ring.

**Independent Switching Processor**

Some routers provide a separate switching processor, independent of the process used for process switching. The switching processor may use Fast Switching using a cache of previously used forwarding entries.

An alternate to fast switching is to down-load a compressed form of the complete forwarding information base (in CISCO routers this is known as CISCO Express Forwarding (CEF)). It is often wise to separate the interface data (chosen interface, link layer address, MTU) from the forwarding data (whether address is reachable, interface entry to use), so that changes in the link layer protocols (such as arp table changes) do not require purging and a complete rebuild of the entire forwarding table.

This approach allows the switch processor to handle large numbers of destination IP addresses, even when there are large numbers of route changes. It therefore scales well to core routers and is used in the high-speed routers of many suppliers. The switch processor does not take part in routing operations, receiving all information from the router's route processor. The forwarding table is therefore pre-compiled by the route processor for each interface card.

Since the route processor down-loads all information, there is no need to forward packets to the process switching task which do not have a forwarding information base entry. These are simply discarded. Switch processors differ in the ability to support multicast, multiple queues, access management (checking rates, authenticating users), etc. An entry is also required for completeness to allow any "special" packets to be sent to the process switching path (e.g. destined for the router itself, to be fragmented, to be authenticated, etc).

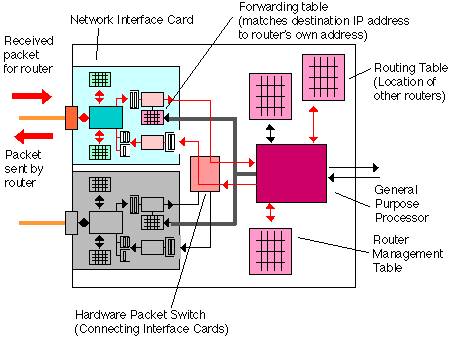
**Multiple Interface Switching Processors**

Some routers provide a number of interface switching processors. Each processor is associated with a group of interface cards and handles all packets received by the group of interface cards. Sometimes this interface switching processor is actually integrated into a single board with the interface cards. A central switching processor must still be allocated to control switching to and from the routing processor (often this is actually the same CPU - they just two different tasks). Communication between interface switching processors still requires the intervention of the main switching processor, and is therefore less optimised than communication between interfaces connected to a common interface switching processor. Network designers may therefore optimise performance by connecting networks which carry related traffic to different interfaces handled by the same interface switching processor. (CISCO call this distributed CEF (dCEF).)

**The Route Processor**

The routing processor is responsible for configuring each network interface card (including the forwarding table), and collecting management information (from the management table associated with each interface ) for the **Router** **Management Table**. The processor is also directly connected to the packet switch which allows the configuration to be (optionally) accessed and modified by establishing a connection form an End-System connected via the network. All configuration data and all performance data collected from the network interface cards are stored in the central router management table. In large routers, a backup routing processor may be provided, in case the main processor fails.

The route processor also processes all packets destined to the router itself (see below).



*Sending and receiving packets which have address which match the router's own IP address. These packets may simply be ICMP echo request messages (to check the router is working) or may be packets directed to the router itself (e.g. indicating routing information from other routers or accessing the information held in the router management table).*

**Packets Destined for the Router (Route Processor)**

When the router responds to received packets (with one of the router's own IP addresses) it will behave as an [End-System](https://erg.abdn.ac.uk/users/gorry/course/intro-pages/es-is.html), rather an [Intermediate-System](https://erg.abdn.ac.uk/users/gorry/course/intro-pages/es-is.html) (where it forwards packets between interfaces). The router general purpose processor will also respond to any [ICMP](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/icmp.html) messages it receives (e.g. sent by the "Ping" program), and may generate ICMP error messages when error events are detected (such as a packet received which can not be routed because the address is not known).

The processor first performs any necessary checks on the packet header and will then determine whether the packet should be discarded, logged (in the router management table) or forwarded. The general purpose CPU is also responsible for configuring the **Forwarding Tables** used by the switching process. The router computes the forwarding tables by processing two local tables: The **Routing Table** and the **Filter Table**.

The "**Routing Table**" contains lists of internet [addresses](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip-address.html) and their corresponding location in the network. A router connected to the Internet will need to be able to identify which interface is to be used to reach every other connected end system. Routers near the centre of a network generally have very large routing tables, those nearer the edges have smaller tables.

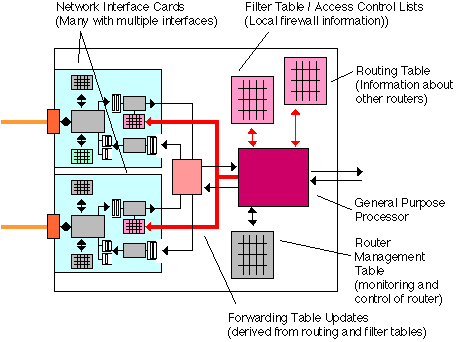
The routing table is constructed by using information supplied when the router is configured (installed) by the manager which it stores in the routing table. Although the routing table may be configured by hand, it is usually configured automatically using a "Routing Protocol". The routing protocol allows routers to periodically (e.g. every few tens of seconds) exchange information about the contents of their own routing tables. After a period of time, the router becomes aware of all the possible ways to reach each end system connected at any point in the network. It therefore adds information to it own routing table about the other routers to which it is connected, building a picture of how to reach other parts of the network. This is achieved by periodically sending packets to all neighboring routers .

The [**filter table**](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/router.html) is usually manually configured, and contains a list of addresses and other packet header details which, if they match a received packet, will cause the packet to be examined in detail and possibly rejected. This may be used to prevent unauthorised packets being forwarded (e.g. to act as a firewall). When a packet is detected by a network interface card (i.e. it matches an entry in the forwarding table), it may be either discarded by the network interface immediately,or forwarded directly to the general purpose processor for further processing. This table is often called an Access Control List (ACL), and may become very complex in some applications.

As the router discovers changes in the routing tables and filter tables it may either

1. invalidate the forwarding cache, causing the fast switching proces sto start build a new forwarding cache.
2. constructs a new forwarding information base for each network interface card, with the appropraite modifications.

This esnures the forwarding engines (switching processors) to be updated. The forwarding information base is optimised for speed (since it needs to be consulted for each received packet), in contrast, the routing table is optimised to ease updating, since received packets received via the routing protocol may require many changes to be made to the routing table (e.g. after the network topology changes following a failure of a communications link).



*As required, network interface cards must be reloaded with new forwarding information which reflects any changes to the network topology. This information is based on the information configured in the filter table and collected from the network and stored in the routing information base.*

*See also:*

[IP](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/ip.html)

[Router Overview](https://erg.abdn.ac.uk/users/gorry/course/inet-pages/router.html)

[Bridge](https://erg.abdn.ac.uk/users/gorry/course/lan-pages/bridge.html)