Packet Routing and Delivery

Although some network connections are point-to-point links between two hosts, most network connections involve more than two hosts, often connected through a hub, switch, router, or other hardware. This means that each host can potentially receive packets intended for other hosts on the network. Thus, each host must be able to recognize whether it is the intended recipient so that it does not incorrectly act upon someone else's packet.

To support this, each packet contains a the link-layer address of its intended recipient. When the host receives a packet intended for another host, it ignores it. In many systems, this filtering is performed in hardware, so that the operating system does not need to concern itself with other hosts' traffic.

> **Important:** Link-layer addressing provides correctness, *not* security. Although the operating system does not send other hosts' traffic to programs through TCP or UDP sockets, it is not obligated to discard those extraneous packets entirely. Packet sniffing tools like `tcpdump` can easily place a network interface into promiscuous mode, allowing those tools to capture and examine packets intended for other hosts on the network.

Because of this design, when two hosts need to communicate across a local physical network, the sending host needs to know the link-layer address of the receiving host before it can start sending packets. To provide that information, the hardware-specific portions of the operating system use various means to convert between a logical address (such as an IP address) and a physical address (such as an Ethernet MAC address). The mechanism for performing this conversion depends on the type of network:

- On Ethernet-like networks, the operating system can obtain a hardware address from an IPv4 address using the *address resolution protocol (ARP)*. An ARP request consists of a broadcast message on the network (a message that all hosts receive) asking who has a particular IP address. The host that owns that IP address then responds with its link-layer address.

  For IPv6 communication over Ethernet, hosts use a similar protocol called the *neighbor discovery protocol (NDP)* (which also serves other purposes, including router discovery) that is based on ICMP.

- Non-Ethernet-like networks use various protocols, but they generally behave similarly at a high level—the network driver (or other software or hardware acting on its behalf) provides a way to map from an IP address to some hardware-specific value that uniquely identifies a packet's destination.
- Point-to-point networks (such as VPN tunnels) do not need to perform this mapping because every packet is being sent to the host on the other end of such a link.
- On cellular networks, when a mobile phone initiates communication with a cellular tower, the tower assigns the phone a specific frequency, time slot, or both. From that point on, the connection behaves as a point-to-point network.

When two hosts need to communicate across a larger distance, however, these protocols are not sufficient for two reasons:

- It would be infeasible to relay a broadcast packet throughout the entire Internet.
- Even if you could obtain the link-layer address of a distant host, that information by itself would not be sufficient to determine how to get packets to that host.

Thus, ARP packets are limited to a physical local area network, and the Internet uses *routing* to determine how to send packets to distant machines. More specifically, a *router* is a type of infrastructure device that knows how to send data from one range of IP addresses to another range.

- *Edge routers*—small routers that provide service to a single customer site—usually know nothing more than the IP range on one side of the network. Any packets sent to the narrow range of IP addresses within your in-home network are sent out through one physical port, and any packets sent to any other address are sent out through the other port (which may then pass through a built-in or external cable modem, DSL modem, or other encoding hardware on its way upstream).

- **Core routers**—routers that provide service for major Internet backbone routes—have multiple physical connections, and keep large routing tables that tell which direction to send packets within a given IP range. They use protocols such as the border gateway protocol (BGP) or the routing information protocol (RIP) to advertise new routes to one another. Whenever a new physical cable comes online, the router at one end tells its peers that it now has a route to the router at the other end. Its peers then propagate that advertisement to other routers, and so on.

Similarly, when a link is severed (whether because of an intentional change or because a backhoe inadvertently severed a fiber bundle while digging for a new sewer main), the routers on either end can detect the problem and reroute packets transparently through a different link. This design allows the Internet as a whole to be dynamically reconfigurable and robust against hardware failures.

Because the network topology is complex and regularly changing, networking performance is also complex and regularly changing. For example, if you have a laptop connected to a Wi-Fi router, and two desktop computers connected by Ethernet, communication between the two desktop computers is faster than communication between the laptop and the desktop computers. Both local connections are probably faster than the connection to any site on the public Internet. And the speed at which you communicate with one site can be much faster or slower than another site simply because the connection passes through different routers, cables, and so on.

The details of routing differ depending on whether you are using IPv4 or IPv6.

IPv4 Routing

An IPv4 address consists of a 32-bit number, divided into a host part and a network part. The host part uniquely identifies a given host on a given physical network. The network part identifies the network to which the host is connected.

> **Note:** The host part of an IP address uniquely identifies a host, but the relationship is not necessarily one-to-one. A host can have multiple interfaces on multiple networks, each with its own IP address. Further, a single network interface can have multiple IP addresses on the same physical network.

Depending on how a particular block of IP addresses has been divided up by its owner, the network part may be as small as 8 bits, or as large as 30 bits.

For networking to work correctly, each host must know three things: its own IP addresses, the IP address of the destination host, and whether that destination host is on the same network as one of its own addresses. The host uses that information to determine whether to send a packet directly to its destination (if the destination is on the same network) or through a router (if it isn't).

Most hosts make this decision using a fairly simple algorithm:

- For IP addresses in which the network bits all match, the host sends the packet directly to the destination host (using ARP or other protocols to discover its physical address).
- For all other IP addresses, the host sends the packet to the router for the primary physical network (using ARP or another protocol to discover the router's physical address based on its IP address). This particular router is known as the *default gateway*.

  If the host knows something special about the IP address—for example, if it knows that the IP address is on the other side of a VPN connection—it may send the packet to a router other than the default gateway, but this is the exception rather than the rule.

To represent how big the network part is, a host uses a *netmask*. A netmask is a 32-bit value in which a 1 bit indicates that the equivalent bit in an IP address is in the network part, and a 0 bit indicates that the matching bit is in the host part. The netmask is usually written in the same format as an IP address. For example, if you have a 28-bit network part, the netmask is `255.255.255.240`. Table 5-1 shows how the IP address for `developer.apple.com` can be decomposed into a network part and a host part based on an example netmask.

Table 5-1 An IPv4 network address and (example) netmask for developer.apple.com

	Network Part	Host Part		
IP address	17.	254.	2.	129
IP address (in binary)	00010001	11111110	00000010	10000001
Netmask	255.	0.	0.	0
Netmask (in binary)	11111111	00000000	00000000	00000000

Note: Because the size of the network portion of an address is just a policy decision made by the administrators of the network to which that IP address belongs, the example in Table 5-1 shows just one of many possible network masks that could theoretically be used for `developer.apple.com` (and is not the actual netmask).

Because the network part and host part are always contiguous (in practice), this netmask is often shortened to a single number representing the number of 1 bits in the netmask, which is usually preceded by a slash. Thus, the aforementioned network with a 28-bit network part is sometimes referred to as a `/28` network, the example in Table 5-1 is a `/8` network, and so on. In theory, you can create a network with any arbitrary combination of zeros and ones in the netmask, but there is no guarantee that all operating systems or network management software will correctly support such a nonstandard configuration.

Given a particular value for the network part, the valid range of IP addresses available for use in the host part is referred to as a *subnet* or *netblock*. A subnet might, for example, use 24 bits for the network part, leaving 8 bits for the host part. This gives you a block of 256 IP addresses to work with.

However, not all of those 256 IP addresses are actually available. Within any given subnet, there are three special IP addresses reserved: the broadcast address, the network address, and the router address.

- The *broadcast address* is an address in which the host part is all ones. If you send a packet to this address, it is received by every host within the same broadcast domain (usually the subnet), and it is never routed beyond the local area network.

  Performance Note: As a general rule, you should avoid sending data to a broadcast address. Instead, use a multicast address. Multicast allows interested hosts to listen without inundating uninterested hosts, and can be set up to work across subnet boundaries.

- The *network address* is an address in which the host part is all zeros. It was used by older operating systems as the broadcast address, so for historical compatibility reasons, this number is reserved.
- The *router address* is the address of your router. It can be any IP address within the subnet (or in rare situations, outside the subnet), but it must be reachable without routing, which means that it must be on the same physical network. Usually, the router address consists of a host part that is all zeros except for the lowest bit (which is one higher than the network address).

Note: As a special exception, point-to-point networks (networks between two hosts in which one of the two hosts acts as a router) do not need a separate router or network address.

In addition, IPv4 reserves certain addresses for specific uses. The design allows you to recognize specific address types by looking for specific patterns in the high-order bits, as listed below:

Address type	IPv4 mask
Unspecified address Used to indicate the absence of an address. May not be assigned to any host.	`0.0.0.0`
Loopback address An address that allows a host to connect back to itself (localhost).	`127.0.0.1`
Multicast address An address used to send packets to any interested party.	`224.0.0.0/4`
Link-local unicast address An address that should never be routed.	`169.254.0.0/16`

Site-local unicast address
An address that should be routed only within the customer site.
	`10.0.0.0/8`
	`172.16.0.0/12`
	`192.168.0.0/16`

IPv6 Routing

An IPv6 address is divided into the following parts:

- A 64-bit network identifier, which in turn is divided into the following parts:
  - Global routing prefix—Tells which service provider owns the number. This part may be organized hierarchically, and thus may effectively be divided into many smaller parts, but this is a policy decision by the service provider, and is not visible outside the organization.
  - Subnet ID—Identifies an individual physical network at the customer site. This part may be organized arbitrarily by the customer site.

  The division between the two parts is arbitrary. An ISP assigning a block to a big company might assign a `/48` block (48-bit global routing prefix, 16-bit subnet, 64-bit interface), allowing the customer to create up to 2^{16} distinct networks. On the other hand, an ISP assigning a block to an individual home might assign a `/64` block in which the ISP owns the entire network part and the customer has only a single network.
- A 64-bit interface identifier that identifies the host within that network. (The host address is often generated programmatically from the host's MAC address.)

Table 5-2 The structure of an IPv6 address

Network part		Host part
Global routing prefix n bits	Subnet ID 64-n bits	Interface ID 64 bits

Note: As with IPv4, the host part of an IP address uniquely identifies a host, but the relationship is almost never one-to-one. A single network interface typically has multiple IP addresses on the

same physical network (as described in Neighbor Discovery and IPv6 Address Assignment). A host can also have multiple interfaces on multiple networks, each with its own IP address.

In concept, IPv6 routing is similar to IPv4 routing. However, there are no reserved broadcast or network addresses. Instead, a special link-local "all nodes" multicast group (`ff02::1`) provides similar functionality. Similarly, there are no variably-sized subnets; the per-interface portion is always 64 bits.

IPv6 reserves certain addresses for specific uses. The design allows you to recognize specific address types by looking for specific patterns in the high-order bits, as listed below:

Address Type	IPv6 mask	Mask bit pattern
Unspecified address Used to indicate the absence of an address. May not be assigned to any host.	`::/128`	`00000000....00000000`
Loopback address An address that allows a host to connect back to itself (localhost).	`::1/128`	`00000000....00000001`
Multicast address An address used to send packets to any interested party.	`FF00::/8`	`11111111`
Link-local unicast address An address that should never be routed.	`FE80::/10`	`11111110 10`
Site-local unicast address An address that should be routed only within the customer site.	`FEC0::/10`	`11111110 11`

All other address patterns are assumed to be globally routable unicast addresses.

Firewalls and Network Address Translation

Most routers are configured to be as transparent as possible from the perspective of hosts on either side. Routers that are not transparent are called firewalls.

A *firewall* is any router that inspects traffic, modifies traffic, or blocks specific subsets of traffic that flows through it. Like a physical firewall (a fireproof wall that prevents fire from spreading from one part of a building to another), network firewalls are commonly used as edge routers to improve the security of corporate or home networks by limiting the ways that an outside attacker can interact with that network.

Firewalls are commonly used to:

- Block specific ports. For example, many firewalls block ports related to file sharing protocols such as SMB or AFP so that those services are accessible only within the local network.
- Block malformed or spoofed network packets. Malformed packets have been used over the years for a number of denial-of-service attacks, including Smurf attacks, the ping of death, and the INVITE of death.
- Perform deep packet inspection to detect and report suspicious traffic.
- Perform *network address translation (NAT)*, in which the firewall changes each packet's source or destination address or port.

Network address translation deserves further explanation. NAT is most commonly used to make traffic from every machine on one side of a firewall appear to have originated from a different IP address. It can serve different purposes depending on how it is configured:

- Masquerading: A configuration in which connections from inside the firewall are modified so that packets appear to have originated at the firewall. When a host inside the firewall makes a connection to a host on the public Internet, the firewall creates a temporary translation rule that rewrites the source address to that of the firewall, and the source port number to a high-numbered port.

  In this configuration, hosts behind the firewall can connect to the Internet, but hosts on the Internet at large cannot make connections back to machines on the inside of the firewall. This significantly improves security, but can break certain network protocols that rely on that ability.

- Destination NAT: A configuration in which connections from outside the firewall result in connections to hosts inside the firewall whose IP addresses are different. This arrangement is often used for certain types of load balancing, to make multiple servers appear as though they were a single server.

Some masquerading firewalls also support protocols that let applications running on hosts inside the firewall request temporary destination NAT rules so that outside hosts can temporarily reach them. The two most common protocols for supporting this are the network address translation port mapping protocol (NAT-PMP) and the internet gateway device (IGD) standardized device control protocol (part of the Universal Plug And Play standard, or UPnP).

In OS X and iOS, Bonjour provides built-in support for creating port mappings through firewalls that support NAT-PMP or UPnP. Services advertised using wide-area Bonjour are automatically mapped. For services advertised in other ways, you can call `DNSServiceNATPortMappingCreate` to create the mapping, and `DNSServiceRefDeallocate` to destroy the mapping. These mappings are also torn down automatically when the process that created the mappings exits.
