the standard, implementers are free to use whatever means are available to build their protocol stack. This means that any given layer from manufacturer A can operate with the same layer from manufacturer B (assuming the relevant specifications are implemented and interpreted correctly).

Here is a brief outline of the seven-layer OSI networking model:

Layer	Name	Description
7	Application	The ***Application Layer*** is the layer that most network users are exposed to, and is the level at which human communication happens. HTTP, FTP, and SMTP are all application layer protocols. The human sits above this layer, interacting with the application.
6	Presentation	The ***Presentation Layer*** deals with data representation, before it reaches the application. This would include MIME encoding, data compression, formatting checks, byte ordering, etc.
5	Session	The ***Session Layer*** manages the logical communications session between applications. NetBIOS and RPC are two examples of a layer five protocol.
4	Transport	The ***Transport Layer*** provides a method of reaching a particular service on a given network node. Examples of protocols that operate at this layer are TCP and UDP. Some protocols at the transport layer (such as TCP) ensure that all of the data has arrived at the destination, and is reassembled and delivered to the next layer in the proper order. UDP is a "connectionless" protocol commonly used for video and audio streaming.
3	Network	IP (the Internet Protocol) is the most common ***Network Layer*** protocol. This is the layer where routing occurs. Packets can leave the link local network and be retransmitted on other networks. Routers perform this function on a network by having at least two network interfaces, one on each of the networks to be interconnected. Nodes on the Internet are reached by their globally unique IP address. Another critical Network Layer protocol is ICMP, which is a special protocol which provides various management messages needed for correct operation of IP. This layer is also sometimes referred to as the ***Internet Layer***.

Layer	Name	Description
2	Data Link	Whenever two or more nodes share the same physical medium (for example, several computers plugged into a hub, or a room full of wireless devices all using the same radio channel) they use the ***Data Link Layer*** to communicate. Common examples of data link protocols are Ethernet, Token Ring, ATM, and the wireless networking protocols (802.11a/b/g). Communication on this layer is said to be link-local, since all nodes connected at this layer communicate with each other directly. This layer is sometimes known as the ***Media Access Control*** (***MAC***) layer. On networks modeled after Ethernet, nodes are referred to by their ***MAC address.*** This is a unique 48 bit number assigned to every networking device when it is manufactured.
1	Physical	The ***Physical Layer*** is the lowest layer in the OSI model, and refers to the actual physical medium over which communications take place. This can be a copper CAT5 cable, a fiber optic bundle, radio waves, or just about any other medium capable of transmitting signals. Cut wires, broken fiber, and RF interference are all physical layer problems.

The layers in this model are numbered one through seven, with seven at the top. This is meant to reinforce the idea that each layer builds upon, and depends upon, the layers below. Imagine the OSI model as a building, with the foundation at layer one, the next layers as successive floors, and the roof at layer seven. If you remove any single layer, the building will not stand. Similarly, if the fourth floor is on fire, then nobody can pass through it in either direction.

The first three layers (Physical, Data Link, and Network) all happen "on the network." That is, activity at these layers is determined by the configuration of cables, switches, routers, and similar devices. A network switch can only distribute packets by using MAC addresses, so it need only implement layers one and two. A simple router can route packets using only their IP addresses, so it need implement only layers one through three. A web server or a laptop computer runs applications, so it must implement all seven layers. Some advanced routers may implement layer four and above, to allow them to make decisions based on the higher-level information content in a packet, such as the name of a website, or the attachments of an email.

The OSI model is internationally recognized, and is widely regarded as the complete and definitive network model. It provides a framework for manufacturers and network protocol implementers that can be used to build networking devices that interoperate in just about any part of the world.

From the perspective of a network engineer or troubleshooter, the OSI model can seem needlessly complex. In particular, people who build and troubleshoot TCP/IP networks rarely need to deal with problems at the Session or Presentation layers. For the majority of Internet network implementations, the OSI model can be simplified into a smaller collection of five layers.

The TCP/IP model

Unlike the OSI model, the TCP/IP model is not an international standard and its definitions vary. Nevertheless, it is often used as a pragmatic model for understanding and troubleshooting Internet networks. The vast majority of the Internet uses TCP/IP, and so we can make some assumptions about networks that make them easier to understand. The TCP/IP model of networking describes the following five layers:

Layer	Name
5	Application
4	Transport
3	Internet
2	Data Link
1	Physical

In terms of the OSI model, layers five through seven are rolled into the topmost layer (the Application layer). The first four layers in both models are identical. Many network engineers think of everything above layer four as "just data" that varies from application to application. Since the first three layers are interoperable between virtually all manufacturers' equipment, and layer four works between all hosts using TCP/IP, and everything above layer four tends to apply to specific applications, this simplified model works well when building and troubleshooting TCP/IP networks. We will use the TCP/IP model when discussing networks in this book.

The TCP/IP model can be compared to a person delivering a letter to a downtown office building. The person first needs to interact with the road itself (the Physical layer), pay attention to other traffic on the road (the Data Link layer), turn at the proper place to connect to other roads and arrive at the correct address (the Internet layer), go to the proper floor and room number (the Transport layer), and finally give it to a receptionist who can take the letter from there (the Application layer). Once they have delivered the message to the receptionist, the delivery person is free to go on their way.

The five layers can be easily remembered by using the mnemonic "**P**lease **D**on't **L**ook **I**n **T**he **A**ttic," which of course stands for "**P**hysical / **D**ata **L**ink / **I**nternet / **T**ransport / **A**pplication."

The Internet protocols

TCP/IP is the protocol stack most commonly used on the global Internet. The acronym stands for ***Transmission Control Protocol*** (***TCP***) and ***Internet Protocol*** (***IP***), but actually refers to a whole family of related communications protocols. TCP/IP is also called the ***Internet protocol suite***, and it operates at layers three and four of the TCP/IP model.

In this discussion, we will focus on version four of the IP protocol (IPv4) as this is now the most widely deployed protocol on the Internet.

IP Addressing

In an IPv4 network, the address is a 32-bit number, normally written as four 8-bit numbers expressed in decimal form and separated by periods. Examples of IP addresses are 10.0.17.1, 192.168.1.1, or 172.16.5.23.

If you enumerated every possible IP address, they would range from 0.0.0.0 to 255.255.255.255. This yields a total of more than four billion possible IP addresses (255 x 255 x 255 x 255 = 4,228,250,625); although many of these are reserved for special purposes and should not be assigned to hosts. Each of the usable IP addresses is a unique identifier that distinguishes one network node from another.

Interconnected networks must agree on an IP addressing plan. IP addresses must be unique and generally cannot be used in different places on the Internet at the same time; otherwise, routers would not know how best to route packets to them.

IP addresses are allocated by a central numbering authority that provides a consistent and coherent numbering method. This ensures that duplicate addresses are not used by different networks. The authority assigns large blocks of consecutive addresses to smaller authorities, who in turn assign smaller consecutive blocks within these ranges to other authorities, or to their customers. These groups of addresses are called sub-networks, or ***subnets*** for short. Large subnets can be further subdivided into smaller subnets. A group of related addresses is referred to as an ***address space***.

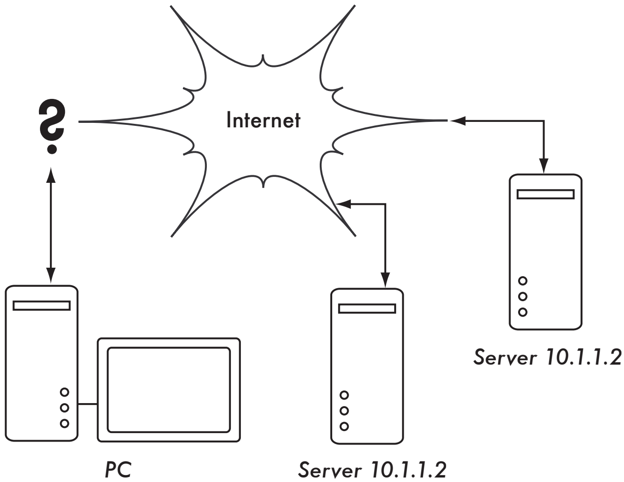


Figure 3.3: Without unique IP addresses, unambiguous global routing is impossible. If the PC requests a web page from 10.1.1.2, which server will it reach?

Subnets

By applying a ***subnet mask*** (also called a ***network mask***, or simply ***netmask***) to an IP address, you can logically define both a host and the network to which it belongs. Traditionally, subnet masks are expressed using dotted decimal form, much like an IP address. For example, 255.255.255.0 is one common netmask. You will find this notation used when configuring network interfaces, creating routes, etc. However, subnet masks are more succinctly expressed using ***CIDR notation***, which simply enumerates the number of bits in the mask after a forward slash (/). Thus, 255.255.255.0 can be simplified as /24. CIDR is short for ***Classless Inter-Domain Routing***, and is defined in RFC1518[1].

A subnet mask determines the size of a given network. Using a /24 netmask, 8 bits are reserved for hosts (32 bits total - 24 bits of netmask = 8 bits for hosts). This yields up to 256 possible host addresses ($2^8 = 256$). By convention, the first value is taken as the ***network address*** (.0 or 00000000), and the last value is taken as the ***broadcast address*** (.255 or 11111111). This leaves 254 addresses available for hosts on this network.

Subnet masks work by applying AND logic to the 32 bit IP number. In binary notation, the "1" bits in the mask indicate the network address portion, and "0" bits indicate the host address portion. A logical AND is performed by comparing two bits. The result is "1" if both of the bits being compared are

---

1. RFC is short for Request For Comments. RFCs are a numbered series of documents published by the Internet Society that document ideas and concepts related to Internet technologies. Not all RFCs are actual standards. RFCs can be viewed online at http://rfc.net/

also "1". Otherwise the result is "0". Here are all of the possible outcomes of a binary AND comparison between two bits.

Bit 1	Bit 2	Result
0	0	0
0	1	0
1	0	0
1	1	1

To understand how a netmask is applied to an IP address, first convert everything to binary. The netmask 255.255.255.0 in binary contains twenty-four "1" bits:

```
        255      255      255      0
        11111111.11111111.11111111.00000000
```

When this netmask is combined with the IP address 10.10.10.10, we can apply a logical AND to each of the bits to determine the network address.

```
  10.10.10.10: 00001010.00001010.00001010.00001010
255.255.255.0: 11111111.11111111.11111111.00000000
        -----------------------------------
   10.10.10.0: 00001010.00001010.00001010.00000000
```

This results in the network 10.10.10.0/24. This network consists of the hosts 10.10.10.1 through 10.10.10.254, with 10.10.10.0 as the network address and 10.10.10.255 as the broadcast address.

Subnet masks are not limited to entire octets. One can also specify subnet masks like 255.254.0.0 (or /15 CIDR). This is a large block, containing 131,072 addresses, from 10.0.0.0 to 10.1.255.255. It could be further subdivided, for example into 512 subnets of 256 addresses each. The first one would be 10.0.0.0-10.0.0.255, then 10.0.1.0-10.0.1.255, and so on up to 10.1.255.0-10.1.255.255. Alternatively, it could be subdivided into 2 blocks of 65,536 addresses, or 8192 blocks of 16 addresses, or in many other ways. It could even be subdivided into a mixture of different block sizes, as long as none of them overlap, and each is a valid subnet whose size is a power of two.

While many netmasks are possible, common netmasks include:

CIDR	Decimal	# of Hosts
/30	255.255.255.252	4
/29	255.255.255.248	8
/28	255.255.255.240	16
/27	255.255.255.224	32
/26	255.255.255.192	64
/25	255.255.255.128	128
/24	255.255.255.0	256
/16	255.255.0.0	65 536
/8	255.0.0.0	16 777 216

With each reduction in the CIDR value the IP space is doubled. Remember that two IP addresses within each network are always reserved for the network and broadcast addresses.

There are three common netmasks that have special names. A /8 network (with a netmask of 255.0.0.0) defines a ***Class A*** network. A /16 (255.255.0.0) is a ***Class B***, and a /24 (255.255.255.0) is called a ***Class C***. These names were around long before CIDR notation, but are still often used for historical reasons.

Global IP Addresses

Have you ever wondered who controls the allocation of IP space? ***Globally routable IP addresses*** are assigned and distributed by ***Regional Internet Registrars*** (***RIR***s) to ISPs. The ISP then allocates smaller IP blocks to their clients as required. Virtually all Internet users obtain their IP addresses from an ISP.

The 4 billion available IP addresses are administered by the ***Internet Assigned Numbers Authority*** (***IANA***, *http://www.iana.org/*). IANA has divided this space into large subnets, usually /8 subnets with 16 million addresses each. These subnets are delegated to one of the five regional Internet registries (RIRs), which are given authority over large geographic areas.

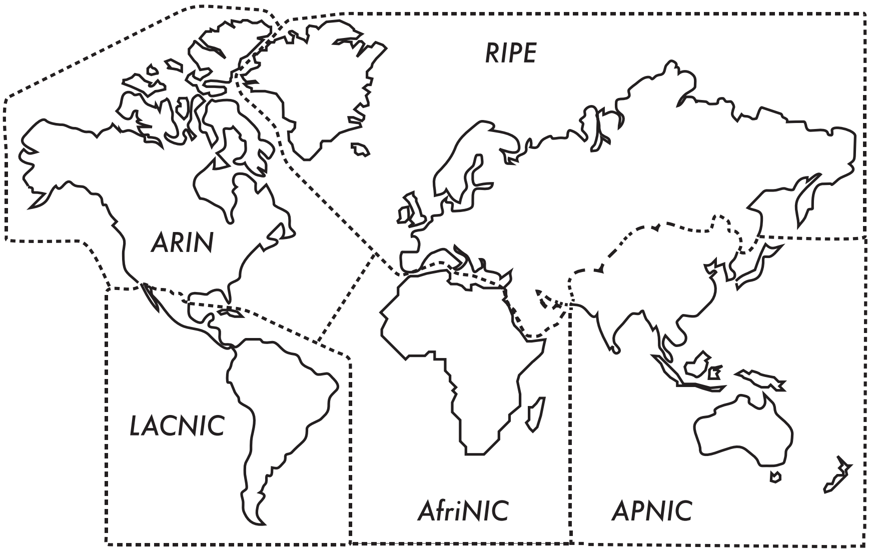


Figure 3.4: Authority for Internet IP address assignments is delegated to the five Regional Internet Registrars.

The five RIRs are:

- African Network Information Centre (AfriNIC, *http://www.afrinic.net/*)
- Asia Pacific Network Information Centre (APNIC, *http://www.apnic.net/*)
- American Registry for Internet Numbers (ARIN, *http://www.arin.net/*)
- Regional Latin-American and Caribbean IP Address Registry (LACNIC, *http://www.lacnic.net/*)
- Réseaux IP Européens (RIPE NCC, *http://www.ripe.net/*)

Your ISP will assign globally routable IP address space to you from the pool allocated to it by your RIR. The registry system assures that IP addresses are not reused in any part of the network anywhere in the world.

Once IP address assignments have been agreed upon, it is possible to pass packets between networks and participate in the global Internet. The process of moving packets between networks is called ***routing***.

Static IP Addresses

A static IP address is an address assignment that never changes. Static IP addresses are important because servers using these addresses may have DNS mappings pointed towards them, and typically serve information to other machines (such as email services, web servers, etc.).

Blocks of static IP addresses may be assigned by your ISP, either by request or automatically depending on your means of connection to the Internet.

Dynamic IP Addresses

Dynamic IP addresses are assigned by an ISP for non-permanent nodes connecting to the Internet, such as a home computer which is on a dial-up connection.

Dynamic IP addresses can be assigned automatically using the ***Dynamic Host Configuration Protocol*** (***DHCP***), or the ***Point-to-Point Protocol*** (***PPP***), depending on the type of Internet connection. A node using DHCP first requests an IP address assignment from the network, and automatically configures its network interface. IP addresses can be assigned randomly from a pool by your ISP, or might be assigned according to a policy. IP addresses assigned by DHCP are valid for a specified time (called the ***lease time***). The node must renew the DHCP lease before the lease time expires. Upon renewal, the node may receive the same IP address or a different one from the pool of available addresses.

Dynamic addresses are popular with Internet service providers, because it enables them to use fewer IP addresses than their total number of customers. They only need an address for each customer who is **active at any one time**. Globally routable IP addresses cost money, and some authorities that specialize in the assignment of addresses (such as RIPE, the European RIR) are very strict on IP address usage for ISP's. Assigning addresses dynamically allows ISPs to save money, and they will often charge extra to provide a static IP address to their customers.

Private IP addresses

Most private networks do not require the allocation of globally routable, public IP addresses for every computer in the organization. In particular, computers which are not public servers do not need to be addressable from the public Internet. Organizations typically use IP addresses from the ***private address space*** for machines on the internal network.

There are currently three blocks of private address space reserved by IANA: 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16. These are defined in RFC1918. These addresses are not intended to be routed on the Internet, and are typically unique only within an organization or group of organizations which choose to follow the same numbering scheme.

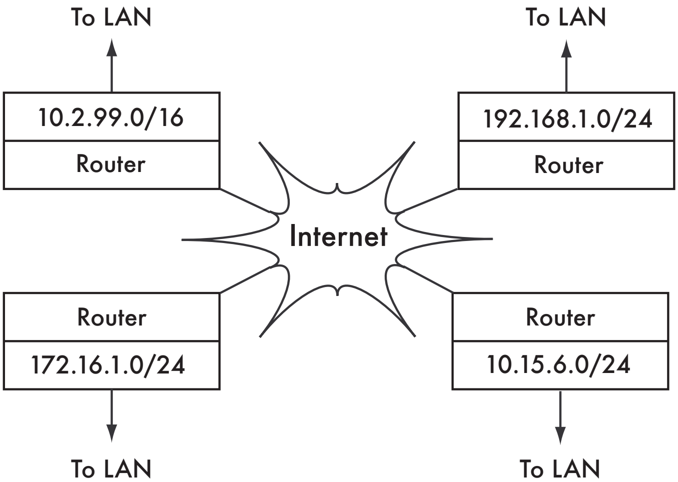


Figure 3.5: RFC1918 private addresses may be used within an organization, and are not routed on the global Internet.

If you ever intend to link together private networks that use RFC1918 address space, be sure to use unique addresses throughout all of the networks. For example, you might break the 10.0.0.0/8 address space into multiple Class B networks (10.1.0.0/16, 10.2.0.0/16, etc.). One block could be assigned to each network according to its physical location (the campus main branch, field office one, field office two, dormitories, and so forth). The network administrators at each location can then break the network down further into multiple Class C networks (10.1.1.0/24, 10.1.2.0/24, etc.) or into blocks of any other logical size. In the future, should the networks ever be linked (either by a physical connection, wireless link, or VPN), then all of the machines will be reachable from any point in the network without having to renumber network devices.

Some Internet providers may allocate private addresses like these instead of public addresses to their customers, although this has serious disadvantages. Since these addresses cannot be routed over the Internet, computers which use them are not really "part" of the Internet, and are not directly reachable from it. In order to allow them to communicate with the Internet, their private addresses must be translated to public addresses. This translation process is known as ***Network Address Translation*** (***NAT***), and is normally performed at the gateway between the private network and the Internet. We will look at NAT in more detail on **Page 43**.

Routing

Imagine a network with three hosts: A, B, and C. They use the corresponding IP addresses 192.168.1.1, 192.168.1.2 and 192.168.1.3. These hosts are part of a /24 network (their network mask is 255.255.255.0).

For two hosts to communicate on a local network, they must determine each others' MAC addresses. It is possible to manually configure each host with a mapping table from IP address to MAC address, but normally the ***Address Resolution Protocol*** (***ARP***) is used to determine this automatically.

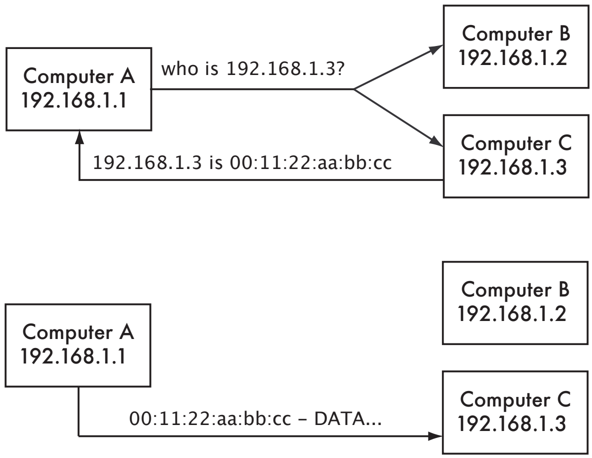


Figure 3.6: Computer A needs to send data to 192.168.1.3. But it must first ask the whole network for the MAC address that responds to 192.168.1.3.

When using ARP, host A broadcasts to all hosts the question, "Who has the MAC address for the IP 192.168.1.3?" When host C sees an ARP request for its own IP address, it replies with its MAC address.

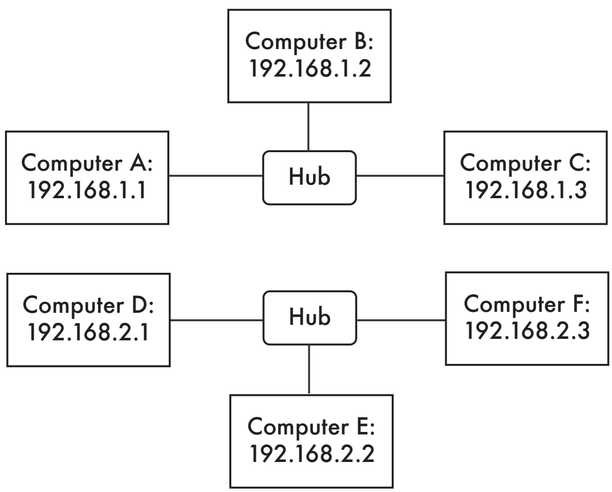


Figure 3.7: Two separate IP networks.

Consider now another network with 3 hosts, D, E, and F, with the corresponding IP addresses 192.168.2.1, 192.168.2.2, and 192.168.2.3. This is another /24 network, but it is not in the same range as the network above. All three hosts can reach each other directly (first using ARP to resolve the IP address into a MAC address, and then sending packets to that MAC address).

Now we will add host G. This host has two network cards, with one plugged into each network. The first network card uses the IP address 192.168.1.4, and the other uses 192.168.2.4. Host G is now link-local to both networks, and can route packets between them.

But what if hosts A, B, and C want to reach hosts D, E, and F? They will need to add a route to the other network via host G. For example, hosts A-C would add a route via 192.168.1.4. In Linux, this can be accomplished with the following command:

```
# ip route add 192.168.2.0/24 via 192.168.1.4
```

...and hosts D-F would add the following:

```
# ip route add 192.168.1.0/24 via 192.168.2.4
```

The result is shown in **Figure 3.8**. Notice that the route is added via the IP address on host G that is link-local to the respective network. Host A could not add a route via 192.168.2.4, even though it is the same physical machine as 192.168.1.4 (host G), since that IP is not link-local.

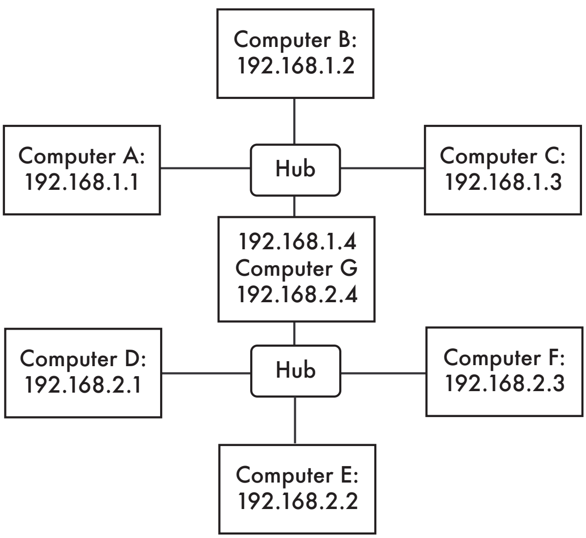
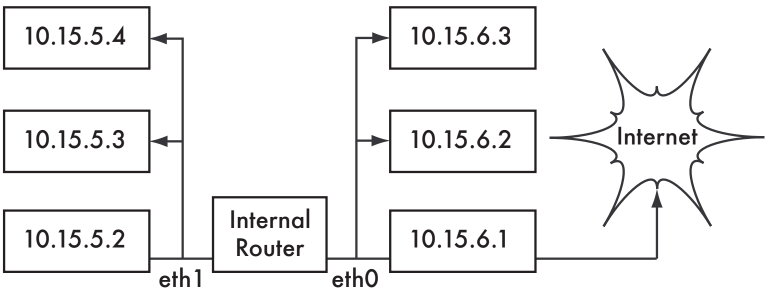


Figure 3.8: Host G acts as a router between the two networks.

A route tells the OS that the desired network doesn't lie on the immediate link-local network, and it must ***forward*** the traffic through the specified router. If host A wants to send a packet to host F, it would first send it to host G. Host G would then look up host F in its routing table, and see that it has a direct connection to host F's network. Finally, host G would resolve the hardware (MAC) address of host F and forward the packet to it.

This is a very simple routing example, where the destination is only a single ***hop*** away from the source. As networks get more complex, many hops may need to be traversed to reach the ultimate destination. Since it isn't practical for every machine on the Internet to know the route to every other, we make use of a routing entry known as the ***default route*** (also known as the ***default gateway***). When a router receives a packet destined for a network for which it has no explicit route, the packet is forwarded to its default gateway.

The default gateway is typically the best route out of your network, usually in the direction of your ISP. An example of a router that uses a default gateway is shown in **Figure 3.9**.




```
Routing table for internal router:

Destination      Gateway      Genmask          Flags  Metric  Iface
10.15.5.0        *            255.255.255.0    U      0       eth1
10.15.6.0        *            255.255.255.0    U      0       eth0
default          10.15.6.1    0.0.0.0          UG     0       eth0
```

Figure 3.9: When no explicit route exists to a particular destination, a host uses the default gateway entry in its routing table.

Routes can be updated manually, or can dynamically react to network outages and other events. Some examples of popular dynamic routing protocols are RIP, OSPF, BGP, and OLSR. Configuring dynamic routing is beyond the scope of this book, but for further reading on the subject, see the resources in **Appendix A**.

Network Address Translation (NAT)

In order to reach hosts on the Internet, RFC1918 addresses must be converted to global, publicly routable IP addresses. This is achieved using a technique known as ***Network Address Translation***, or ***NAT***. A NAT device is a router that manipulates the addresses of packets instead of simply forwarding them. On a NAT router, the Internet connection uses one (or more) globally routed IP addresses, while the private network uses an IP address from the RFC1918 private address range. The NAT router allows the global address(es) to be shared with all of the inside users, who all use private addresses. It converts the packets from one form of addressing to the other as the packets pass through it. As far as the network users can tell, they are directly connected to the Internet and require no special software or drivers. They simply use the NAT router as their default gateway, and address packets as they normally would. The NAT router translates outbound packets to use the global IP address as they leave the network, and translates them back again as they are received from the Internet.

The major consequence of using NAT is that machines from the Internet cannot easily reach servers within the organization without setting up explicit forwarding rules on the router. Connections initiated from within the private address space generally have no trouble, although some applications (such as Voice over IP and some VPN software) can have difficulty dealing with NAT.

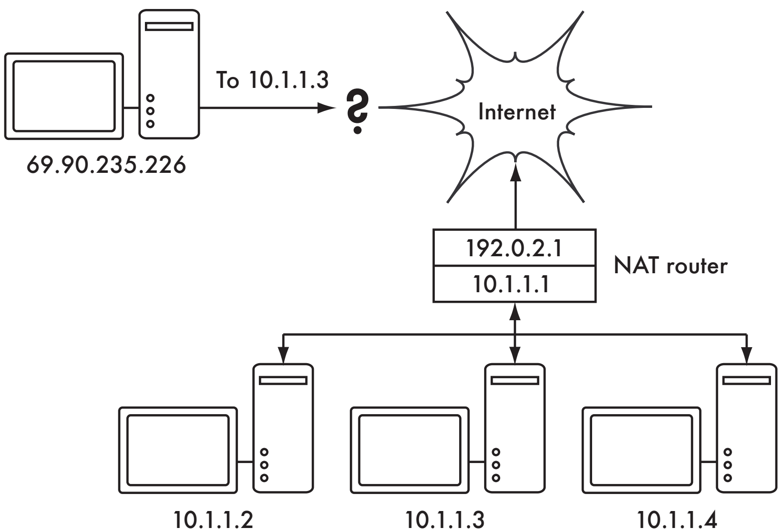


Figure 3.10: Network Address Translation allows you to share a single IP address with many internal hosts, but can make it difficult for some services to work properly.

Depending on your point of view, this can be considered a bug (since it makes it harder to set up two-way communication) or a feature (since it effectively provides a "free" firewall for your entire organization). RFC1918 addresses should be filtered on the edge of your network to prevent accidental or malicious RFC1918 traffic entering or leaving your network. While NAT performs some firewall-like functions, it is not a replacement for a real firewall.

Internet Protocol Suite

Machines on the Internet use the Internet Protocol (IP) to reach each other, even when separated by many intermediary machines. There are a number of protocols that are run in conjunction with IP that provide features as critical to normal operations as IP itself. Every packet specifies a protocol number which identifies the packet as one of these protocols. The most commonly used protocols are the ***Transmission Control Protocol*** (***TCP***, number 6), ***User Datagram Protocol*** (***UDP***, number 17), and the ***Internet Control Message Protocol*** (***ICMP***, number 1). Taken as a group, these protocols (and others) are known as the ***Internet Protocol Suite***, or simply ***TCP/IP*** for short.

The TCP and UDP protocols introduce the concept of port numbers. Port numbers allow multiple services to be run on the same IP address, and still be distinguished from each other. Every packet has a source and destination port number. Some port numbers are well defined standards, used to reach well known services such as email and web servers. For example, web servers normally ***listen*** on TCP port 80, and SMTP email servers listen on TCP port 25. When we say that a service "listens" on a port (such as port 80), we mean that it will accept packets that use its IP as the destination IP address, and 80 as the destination port. Servers usually do not care about the source IP or source port, although sometimes they will use them to establish the identity of the other side. When sending a response to such packets, the server will use its own IP as the source IP, and 80 as the source port.

When a client connects to a service, it may use any source port number on its side which is not already in use, but it must connect to the proper port on the server (e.g. 80 for web, 25 for email). TCP is a ***session oriented*** protocol with guaranteed delivery and transmission control features (such as detection and mitigation of network congestion, retries, packet reordering and reassembly, etc.). UDP is designed for ***connectionless*** streams of information, and does not guarantee delivery at all, or in any particular order.

The ICMP protocol is designed for debugging and maintenance on the Internet. Rather than port numbers, it has ***message types***, which are also numbers. Different message types are used to request a simple response from another computer (echo request), notify the sender of another packet of a possible routing loop (time exceeded), or inform the sender that a packet that could not be delivered due to firewall rules or other problems (destination unreachable).

By now you should have a solid understanding of how computers on the network are addressed, and how information flows on the network between them. Now let's take a brief look at the physical hardware that implements these network protocols.

Ethernet

Ethernet is the name of the most popular standard for connecting together computers on a ***Local Area Network*** (***LAN***). It is sometimes used to connect individual computers to the Internet, via a router, ADSL modem, or wireless device. However, if you connect a single computer to the Internet, you may not use Ethernet at all. The name comes from the physical concept of the ether, the medium which was once supposed to carry light waves through free space. The official standard is called IEEE 802.3.

The most common Ethernet standard is called 100baseT. This defines a data rate of 100 megabits per second, running over twisted pair wires, with modular RJ-45 connectors on the end. The network topology is a star, with switches or hubs at the center of each star, and end nodes (devices and additional switches) at the edges.

MAC addresses

Every device connected to an Ethernet network has a unique MAC address, assigned by the manufacturer of the network card. Its function is like that of an IP address, since it serves as a unique identifier that enables devices to talk to each other. However, the scope of a MAC address is limited to a broadcast domain, which is defined as all the computers connected together by wires, hubs, switches, and bridges, but not crossing routers or Internet gateways. MAC addresses are never used directly on the Internet, and are not transmitted across routers.

Hubs

Ethernet ***hubs*** connect multiple twisted-pair Ethernet devices together. They work at the physical layer (the lowest or first layer). They repeat the signals received by each port out to all of the other ports. Hubs can therefore be considered to be simple repeaters. Due to this design, only one port can successfully transmit at a time. If two devices transmit at the same time, they corrupt each other's transmissions, and both must back off and retransmit their packets later. This is known as a ***collision***, and each host remains responsible for detecting collisions during transmission, and retransmitting its own packets when needed.

When problems such as excessive collisions are detected on a port, some hubs can disconnect (***partition***) that port for a while to limit its impact on the rest of the network. While a port is partitioned, devices attached to it cannot communicate with the rest of the network. Hub-based networks are generally more robust than coaxial Ethernet (also known as 10base2 or ThinNet), where misbehaving devices can disable the entire segment. But hubs are limited in their usefulness, since they can easily become points of congestion on busy networks.

Switches

A ***switch*** is a device which operates much like a hub, but provides a dedicated (or ***switched***) connection between ports. Rather than repeating all traffic on every port, the switch determines which ports are communicating directly and temporarily connects them together. Switches generally provide much better performance than hubs, especially on busy networks with many computers. They are not much more expensive than hubs, and are replacing them in many situations.

Switches work at the data link layer (the second layer), since they interpret and act upon the MAC address in the packets they receive. When a packet arrives at a port on a switch, it makes a note of the source MAC address, which it associates with that port. It stores this information in an internal ***MAC table***. The switch then looks up the destination MAC address in its MAC table, and transmits the packet on the matching port. If the destination MAC address is not found in the MAC table, the packet is then sent to all of the connected interfaces. If the destination port matches the incoming port, the packet is filtered and is not forwarded.

Hubs vs. Switches

Hubs are considered to be fairly unsophisticated devices, since they inefficiently rebroadcast all traffic on every port. This simplicity introduces both a performance penalty and a security issue. Overall performance is slower, since the available bandwidth must be shared between all ports. Since all traffic is seen by all ports, any host on the network can easily monitor all of the network traffic.

Switches create virtual connections between receiving and transmitting ports. This yields better performance because many virtual connections can be made simultaneously. More expensive switches can switch traffic by inspecting packets at higher levels (at the transport or application layer), allow the creation of VLANs, and implement other advanced features.

A hub can be used when repetition of traffic on all ports is desirable; for example, when you want to explicitly allow a monitoring machine to see all of the traffic on the network. Most switches provide ***monitor port*** functionality that enables repeating on an assigned port specifically for this purpose.

Hubs were once cheaper than switches. However, the price of switches have reduced dramatically over the years. Therefore, old network hubs should be replaced whenever possible with new switches.

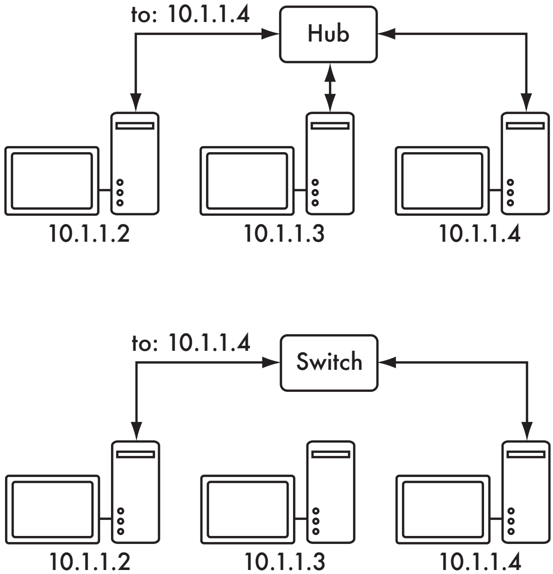


Figure 3.11: A hub simply repeats all traffic on every port, while a switch makes a temporary, dedicated connection between the ports that need to communicate.

Both hubs and switches may offer ***managed*** services. Some of these services include the ability to set the link speed (10baseT, 100baseT, 1000baseT, full or half duplex) per port, enable triggers to watch for network events (such as changes in MAC address or malformed packets), and usually include ***port counters*** for easy bandwidth accounting. A managed switch that provides upload and download byte counts for every physical port can greatly simplify network monitoring. These services are typically available via SNMP, or they may be accessed via telnet, ssh, a web interface, or a custom configuration tool.

Routers and firewalls

While hubs and switches provide connectivity on a local network segment, a router's job is to forward packets between different network segments. A router typically has two or more physical network interfaces. It may include support for different types of network media, such as Ethernet, ATM, DSL, or dial-up. Routers can be dedicated hardware devices (such as Cisco or Juniper routers) or they can be made from a standard PC with multiple network cards and appropriate software.

Routers sit at the ***edge*** of two or more networks. By definition, they have one connection to each network, and as border machines they may take on other responsibilities as well as routing. Many routers have ***firewall*** capabilities that provide a mechanism to filter or redirect packets that do not fit security or access policy requirements. They may also provide Network Address Translation (NAT) services.

Routers vary widely in cost and capabilities. The lowest cost and least flexible are simple, dedicated hardware devices, often with NAT functionality, used to share an Internet connection between a few computers. The next step up is a software router, which consists of an operating system running on a standard PC with multiple network interfaces. Standard operating systems such as Microsoft Windows, Linux, and BSD are all capable of routing, and are much more flexible than the low-cost hardware devices. However, they suffer from the same problems as conventional PCs, with high power consumption, a large number of complex and potentially unreliable parts, and more involved configuration.

The most expensive devices are high-end dedicated hardware routers, made by companies like Cisco and Juniper. They tend to have much better performance, more features, and higher reliability than software routers on PCs. It is also possible to purchase technical support and maintenance contracts for them.

Most modern routers offer mechanisms to monitor and record performance remotely, usually via the Simple Network Management Protocol (SNMP), although the least expensive devices often omit this feature.

Other equipment

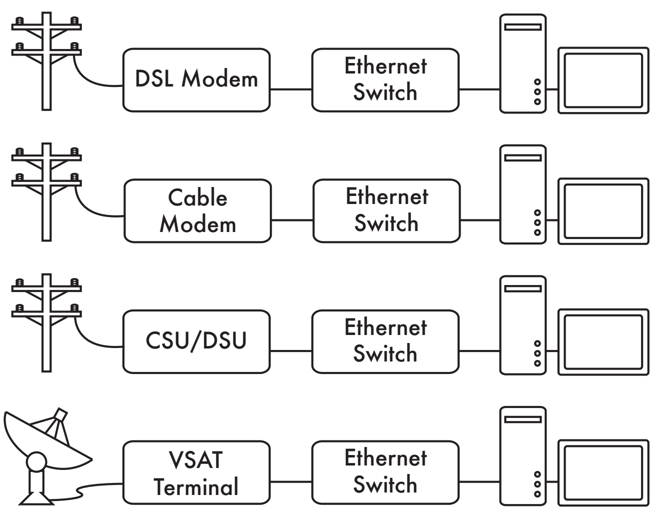


Figure 3.12: Many DSL modems, cable modems, CSU/DSUs, wireless access points, and VSAT terminals terminate at an Ethernet jack.

Each physical network has an associated piece of terminal equipment. For example, VSAT connections consist of a satellite dish connected to a terminal that either plugs into a card inside a PC, or ends at a standard Ethernet connection. DSL lines use a ***DSL modem*** that bridges the telephone line to a local device, either an Ethernet network or a single computer via USB. ***Cable modems*** bridge the television cable to Ethernet, or to an internal PC card bus. Some kinds of telecom circuit (such as a T1 or T3) use a CSU/DSU to bridge the circuit to a serial port or Ethernet. Standard dialup lines use modems to connect a computer to the telephone, usually via a plug-in card or serial port. And there are many different kinds of wireless networking equipment that connect to a variety of radios and antennas, but nearly always end at an Ethernet jack.

The functionality of these devices can vary significantly between manufacturers. Some provide mechanisms for monitoring performance, while others may not. Since your Internet connection ultimately comes from your ISP, you should follow their recommendations when choosing equipment that bridges their network to your Ethernet network.

Putting it all together

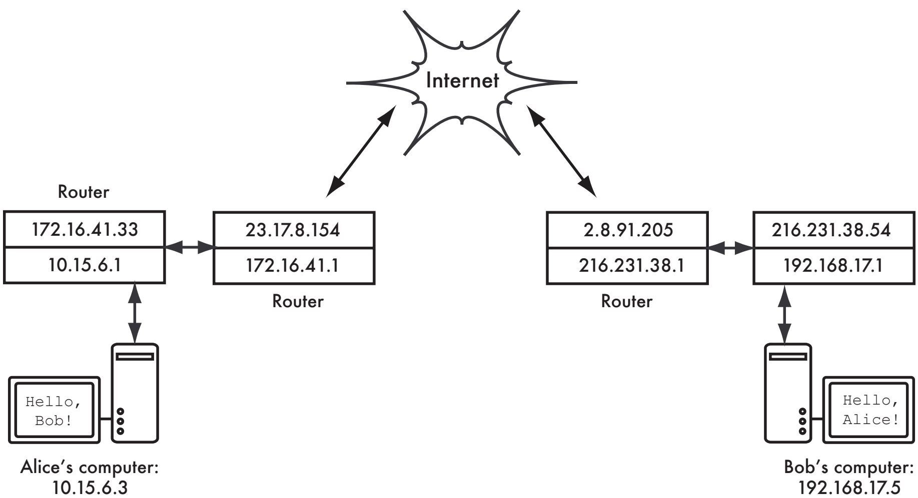


Figure 3.13: Internet networking. Each network segment has a router with two IP addresses, making it "link local" to two different networks. Packets are forwarded between routers until they reach their ultimate destination.

Once all network nodes have an IP address, they can send data packets to the IP address of any other node. Through the use of routing and forwarding, these packets can reach nodes on networks that are not physically connected to the originating node. This process describes much of what "happens" on the Internet.

In this example, you can see the path that the packets take as Alice chats with Bob using an instant messaging service. Each dotted line represents an