The address and Networks

Every computer that sends the data must have a unique identifier. Such an identifier is usually called an address. In some network technologies, the address indicates a specific machine, while in others indicates the point of attachment to the network, which is commonly called the interface. As a result, a single machine on a network that has several interfaces can have several IP addresses, one for each of these interfaces. Interfaces are usually physically distinguishable connections (i.e. the sockets to which the network cable is connected), but they can also be logical connections that have one common physical connection. You may also come across another solution called interface multiplexing, which is used in the ATM network. The ATM logical host division are several groups that allows every single group to be treated as a separate logical network even though all the hosts are attached to one physical network. The equipment connected to this type of physical network can simultaneously belong to several logical networks thanks to establishing several logical connections, each with its own IP address. Machines that have multiple addresses are referred to as multi-homed. All routers are multihomed by definition, as they transport packets between several networks. However, not all machines are referred to as multi-homed routers, e.g. one machine can have several connections to the network and perform the function of a file server shared by several different networks without routing information between these networks.

The IP address structure

IP addresses are 32 bits long. They are considered as a sequence of four bytes or, in other words, four octets (8-bit bytes). To record the IP address, each octet must be converted to decimal notation and separate the resulting four decimal numbers with dots. So a 32-bit IP address:

10101100 00011101 00100000 01000010

Above-mentioned address is saved as:

172.29.32.66

Such format is known as the dotted decimal notation. As it is much more is convenient in general use, we will use it in most cases. However, there will be the cases when it will be more convenient to work with hexadecimal representation of 32-bit addresses, as it will facilitate or better understand some operations. In hexadecimal notation, the IP address shown above will be represented as follows:

0xAC1D2042

Although the IP address is a single 32-bit number, the set of IP addresses is not flat. Instead, addresses built up are based on a two-level hierarchy of networks and hosts that are part of these networks. Each of these two address spaces is identified by a specific portion of an IP address, as a result of which each IP address can be divided into a network number and a host number. In the IP protocol, the network number represents the collection of machines that can communicate directly at the layer of the second ISO-OSI network reference model. This layer is the data link layer that reflects the action solutions such as Ethernet, Token Ring, FDDI (Fiber Distributed Data Interconnect), as well as point-to-point links. Each of these network technologies is treated by IP as one network, whether or not it is one network cable, or it consists of several segments connected by repeaters, bridges, or switches. You shouldn't be surprised to learn that the host number specifies the specific machine that belongs to that particular machine network. Figure 1-1 shows an example of the addressing method described above.

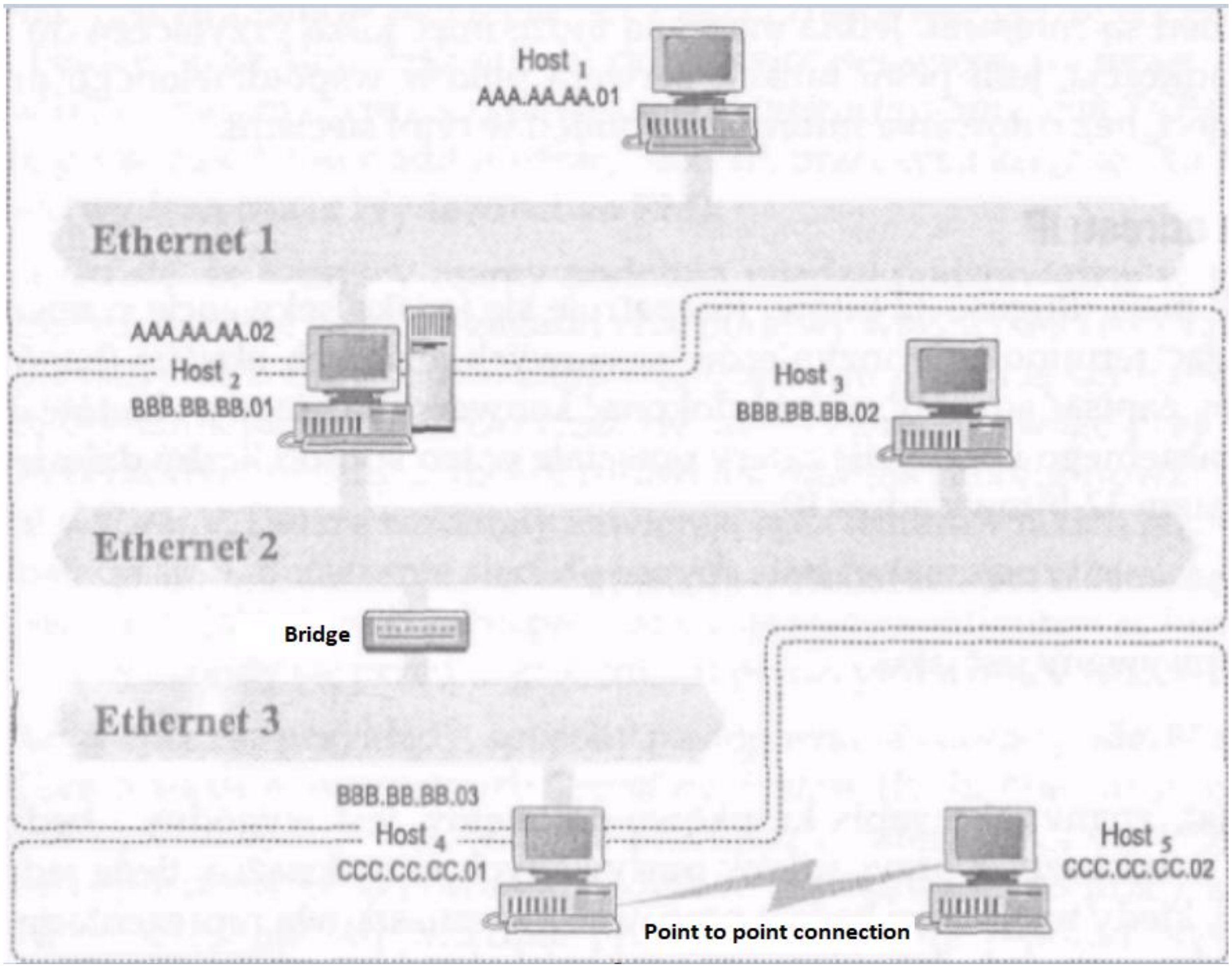


Figure 1-1. Ethernet 2 and 3 are one single network

In Figure 1-1, Ethernet 2 and 3 form a single IP network even though they are separated by a bridge due to the fact that the device such as a bridge is invisible from the protocols of the network layer of IP. Host2, Host3 and Host4 all have IP addresses, which have the same network number assigned to that dual-linked Ethernet system over the bridge. The serial link between Host4 and HostS creates a second IP network and these hosts will have addresses consisting of a number network created by this serial connection. Ethernet 1 is the third IP network and Host1 and Host2 will have IP addresses including its address. Host2 and Host4 have two IP addresses: they are multi-homed and can act as routers.

The two-tier structure of IP addresses will be important later in this document when it comes to routing. For now it suffices to indicate which part of the IP address is the network number and which part is the host number. Putting a network number in an IP address makes the host address dependent on the network on which the host is located. This means that if the host is moved to another network, its address must be changed. Unlike other networking technologies, such as Novell's IPX, where the address is derived from the hardware address of the network card or AppleTalk from Apple Computer, where the address is chosen automatically. The IP is defined and assigned manually. Although there are protocols such as Boot Strap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP), which assists in determining an IP address for a machine on the network, the servers that support these protocols require manual configuration, and not all devices on the network are capable to take advantage of these services. Having to change the host number after changing his workplace means always additional tasks for staff responsible for network maintenance.

Net and mask numbers

All IP addresses consist of a network number and a host number of that network. However, the limit between the network number and the host number can vary. As the router and host software could easily determine where the mentioned address division is located, each of them has attached information in the form of a netmask. Such mask is a 32-bit number (just like an IP address) in which all bit portions that define the network are equal to 1 and the bits specifying the host portion of the address are set to 0. For example:

11111111 11111111 00000000 00000000

means that the first 16 bits of the IP address are associated with the network number, and the last 16 bits represent the host number of that network. The computer can easily calculate the network number from the IP address using bitwise AND operation between an IP address and its mask. Initially, the netmasks may have contained non-contiguous 1 bits. This practice has however been changed, partially because of the difficulties it posed, and in part to simplify the exchange of routing information. Currently, all masks must have all 1 bits adjacent. This means that the following mask:

11111111 11111111 00000011 00000000

it is not allowed because the last two bits of 1 are not adjacent to each other. This limitation did not make any bigger trouble because, until its introduction, few masks were in use where the 1 bits were not contiguous.

Like the IP address, the network mask is traditionally represented using either the dotted decimal notation or hexadecimal. So the mask can be written as 255.255.254.0 or as OxFFFFFE00 - this is the more common way used in software development.

However, since masks are always associated with an IP address, and without it, getting the new mask format becomes more popular. Since it is now required to write as a continuous sequence of bits 1, it is possible to use the concept of a mask, e.g. 23-bit. Such a term unequivocally says that we mean a mask composed of 23 bits of 1s followed by 9 bits of 0s or in the notation hexadecimal OxFFFFFE00. This simplifies the statement that "the network starts with the address 192.168.2.0 from mask 255.255.254.0 " and saves it as 192.168.2.0/23. This new writing of addresses and masks is called writing address/mask.

Display format
192.168.2.0/23
192.168.2.0 255.255.254.0
192.168.2.0 0xFFFFFE00

Table 1-1. Information about mask display format

The basic address/mask notation allows you to easily describe IP addresses from networks of any size, starting with a simple point-to-point link where two hosts in a network work, ending with networks with many million hosts. For example, considering the two addresses shown in Figure 1-2. Since they have the same 23-bitwise prefix and are consecutive numbers, it is possible to write the address space of both mentioned addresses using the aforementioned notation, resulting in an address in the form of 192.168.10.0/23.

192.168.10.0 = 11000000 10101000 00001010 00000000

192.168.11.0 = 11000000 10101000 00001011 00000000

255.255.254.0 = 11111111 11111111 11111110 00000000

Figure 1-2. Two addresses with a common 23-bit prefix.

Not all combinations of addresses and netmasks can be correctly written using this notation. Picture 1-3 shows four addresses cannot be represented by one address/mask entry. It happens so because the addresses, despite their continuity, do not have the same 22-bit prefix. Therefore, it is not possible to specify a 22-bit mask that would cover all these addresses. If you want to write down these addresses by providing 192.168.10.0/22, only two of the four given addresses will be included and the other two will be ignored.

192.168.10.0 = 11000000 10101000 00001010 00000000

192.168.11.0 = 11000000 10101000 00001011 00000000

192.168.12.0 = 11000000 10101000 00001100 00000000

192.168.13.0 = 11000000 10101000 00001101 00000000

255.255.???.0 = 11111111 11111111 11111?00 00000000

Figure 1-3. *Four addresses with no common 22-bit prefix*

Instead of such a record, two separate specifications should be used: 192.168.10.0/23 and 192.168.12.0/23, which means two separate entries in the routing table. Does 192.168.10.0/22 specify any valid address space? Yes and no. If you use a mask with this address, it turns out that the resulting address space is the same as for the address 192.168.8.0/22. Is this type of record is the primary address important? Yes! Even experienced administrators mistakenly believe that the address space thus described is from 192.168.10.0 to 192.168.13.255, although the computer will designate the address space based on the 192 .168.1 0.0/22 from 192.168.8.0 to 192.168.11.255. These are, of course, two completely different address spaces. Such miswriting can cause duplicate address assignments, routing problems, and other mysterious errors.

If you want to avoid this and make the records unambiguous, the base address, masked with the given mask, cannot have any 1 bit in the host numbers part. This limitation is so important that everyone who does a well-written network program will force such a write and inform about an address error in case of failure to comply with this rule. The general rule is as follows: for a number of N base addresses having the same prefix, N must be the base of the power of 2, and the last octet containing the network number (in which no bits are specifying the host number) must be completely divisible by N.

IP address classes

The basic way of writing addresses, described above, allows you to easily distinguish the size of the address part network and the part that determines the number of hosts in this network. You can easily count the hosts on the network, then this number round to the nearest power of two, and on this basis apply for the net number and mask for this net. You should also remember to add the appropriate number of backup addresses that will allow you to expand the network in the future. However, the allocation of network addresses was not always done this way. In the initial period of development of the IP network the masks had a fixed size so that after adding them to network numbers, network classes were created. Although they were replaced with the more flexible architecture described above, references to them. Some routing protocols, such as RIP, still use this concept, so let's focus on basic network classes and their evolution towards the modern network class architecture used today.

The authors of IP did not anticipate that this protocol would have to support a network the size of today's Internet. They assumed that there would be a need to support only a few large networks (operating in large computer companies and major universities), an average number of medium-sized networks, and many small networks. Therefore, they created three network classes: class A intended for the largest networks, class B - for medium-sized networks, and class C - for small networks. They also decided to make routing decisions easier and coded the network class in the first few IP address bits, according to the principle shown in Figure 1-4.

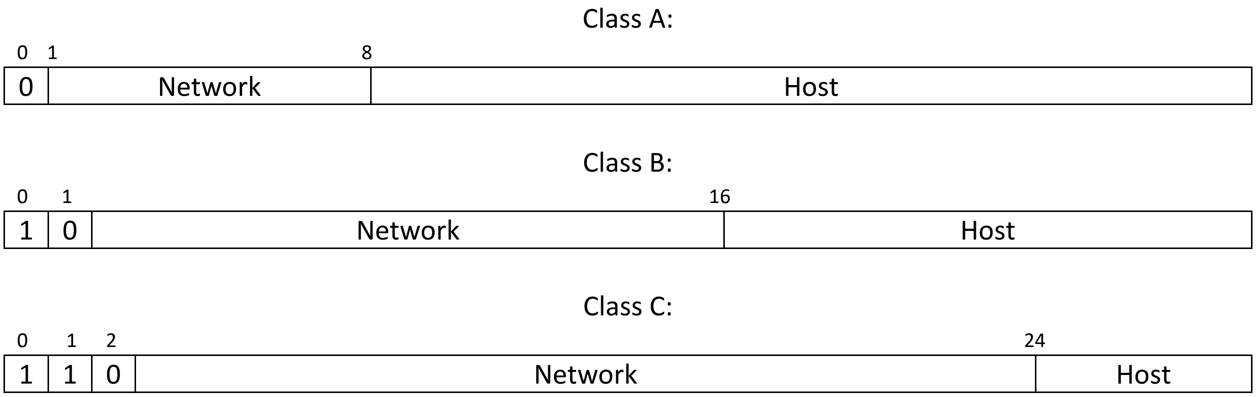


Figure 1-4. The address class is encoded in the first few bits

If the first bit of the address is 0, then the network belongs to class A. In a class A network, the first octet is the network number, the other three octets identify the host on that network. Since the first bit of the address is permanently set to 0, you can use only 127 class A networks, each capable of addressing more than 16 million hosts. If the first two bits of the address are 10, the network is class B. In a class B network, the first two octets are the network number. and the next two - the host number on the network. This allows for the creation of 16,384 class B networks (note that similar to in the previous class, the first two bits are constant), and there can be 65,000 hosts in each of them. Finally, if the first three bits are 110, the network is class C. In a class C network, the first three octets are the network number and the last octet is the host number on the network. This allows for about 2 million networks, each of which can consist of 256 hosts.

Notice how easy it is to determine the network class from the first few bits, and then find the network number part of the address and the host number part. Such simplicity was necessary because computers in the past had much less processing power than they do today.

According to the original definition, addresses, where the first three bits are 111, belong to class D and are intended for future use. Since then, the definition of this class network has changed, and class D is now defined as addresses where the first four bits are 1110. These addresses do not represent a single device, but a set of devices that are part of an IP group called multicast. Addresses starting with 1111 are now called class E addresses and are reserved for future use. Probably, if for the next class of addresses, some way of using them is assigned, the class definition will be modified so that class E will start from 11110, and the new defined class F (as a reserve for the future) will be distinguished by the leading bits of the form 11111.

So how do the network classes described above compare to their newest counterparts? Note that the Class A network has an 8-bit netmask. This means that such a network with the number 10.0.0.0 can be described as 10.0.0.0/8 using classless notation. Also, the natural netmask for a class B network is 16 bits long, and for a class C network, it is 24 bits long. As a result of the mask lengths determined in this way, the designation of the B-class network 172.16.0.0 will be as follows: 172.16.0.0/16, and for the class C network with the address 192.168.1.0 - 192.168.1.0/24. Note, however, that while all networks previously known as Class B networks have 16-bit masks, they are not, it is true that all 16-bit masks are class B networks. Consider the example of 10.0.0.0/16 networks. It uses a 16-bit mask, but still remains a class A network (or rather part of such a network) since its binary representation still starts at bit 0. The network described by 192.168.0.0/16, which is not a network, is similarly constructed. Class B, but a set of 256 Class C networks. These differences are significant when dealing with hosts and protocols that are aware of the existence of network classes. In such cases, the correct configuration of the mask is extremely important for the system operation. When using classless addressing, a 16-bit mask is simply a 16-bit mask.

Subnets and super networks (supernets)

As the makers of IP protocols gained network experience, they found that the initially established network classes allowed networks to be allocated sizes that did not fit the needs of emerging LAN technologies. For example, there is no need to allocate a class B network that can address over 65,000 hosts, an Ethernet network with up to 1,200 devices. A solution called subnetwork partitioning has been developed in which network masks are actually used for the first time.

In IP subnets, the bits belonging to the host's IP address are used as bits to extend the network number. For example, in a class A 10.0.0.0 network, the network number is described by the first 8 bits and the remaining 24 bits make up the host number. The authors of the IP network realized that it is possible to divide this network into subnets by using the next 8 bits of the address, which will be assigned from the host address to the network address, as shown in Figure 1-5. This solution allows creating 256 subnets and 65,000 hosts in each of them. It is also possible to use 16 bits of the host number to determine the subnet addresses, which brings the number of subnets to 65,000 and the number of hosts on each to 256.

Net masks do not have to follow consecutive boundaries defined by 8-bit portions of the IP address. This solution is used in many places because of the way of dividing an address into a network part and a host number is easy to remember. If we do not divide class A 10.0.0.0 networks into subnets, the division between network address and host address is at the first period in the address in decimal. If we use an 8-bit subnet (i.e. a 16-bit netmask), the partition boundary between the subnet and the host address will be at the second dot. If, on the other hand, we use a 16-bit subnet (24-bit network mask), the dividing line will run at the third dot.

Network	Subnetwork	Host
10		27.9.4
10	27	9.4
10	27.9	4

Figure 1-5: Different interpretations of the address 10.27.9.4

Although such facilitations do not matter for computers, they are very convenient for humans and allow to divide the address into individual parts in a more natural way. For example, if we decide to use a 10-bit mask in our sample network 10.0.0.0, we will get 1024 subnets with 4 million hosts on each of them. In this case, the partition line between the subnetwork number and the host number is within the third octet and it is not visible in the dotted decimal notation. Consider the addresses 10.1.190.0 and 10.1.191.1. Are those addresses on the same subnet? Yes, but the address of 10.1.192.1 will no longer be hers. Even the hexadecimal notation of the address does not clearly show this chapter. Only binary notation allows you to clearly distinguish between subnets.

The subnet mask always has at least 1 bits as there are in the natural mask for a given network class. This means that a subnet is always smaller than a network, no matter what class the network comes from. A few years ago, when problems related to the depletion of the address space began, attention was paid to the fact that there is no technical justification for such a rigid treatment of masks. Why not allocate network addresses with masks greater than the natural mask for Class C networks and create blocks of several C networks treated as one network or superlattice? Actually, why limit this approach to Class C networks? Why not combine more class B networks into one super network? Such solutions are the basis of Classless Interdomain Routing between domains. CIDR), which creates the classless architecture used in the network today. By using a netmask to designate both subnets and supernets, a new group of classless routing protocols has emerged, allowing the extension of routing functionality that previously only was possible between class networks. Classless routing protocols and classless protocols cannot be mixed because the latter require knowledge of the address mask, while the class protocol itself determines the mask for the network class based on the first bits of the address. However, it is possible to combine both types of protocols in a controlled manner at the edge of a routing domain. However, such a solution should be used as a last resort and with full awareness of its consequences.

Broadcast and multicast addresses

There are times when a host on an IP network must communicate with all other hosts on that network. Since there is no easy way to tell what other addresses on the network are assigned to hosts, and it is even hard to tell which hosts are currently running, the host can send a copy of the message to each address on the network in turn. It is a waste of network bandwidth and the power of computers running on it. To deal with this problem, IP defines the address 255.255.255.255 as the broadcast address on the local network. Each host working in the IP network receives incoming messages to its IP address and the broadcast address.

Local network broadcast works fine if the host just wants to send the message to other hosts connected directly to the same network. There are, however, situations when a host wants to send a packet to all hosts on a given IP network, but which are not directly connected to its physical network. IP defines such a packet as directed broadcast. Its address contains the number of the network to which it is directed, and all bits of the host number set to 1. So a broadcast for the 10.0.0.0/8 network will have the address 10.255.255.255, and for the network 172.29.0.0/16, it will be the address 172.29.255.255. Due to the potential threat from dishonest network users or ignorant people, many routers can be configured to drop targeted broadcast packets without passing them inside the network they protect. Some versions of older software used bits 0 instead of 1 to denote broadcast addresses. Even though such systems are declining, you may come across them, especially if older systems are running on your network. The latest software should accept both ways of addressing broadcast packets and be able to configure the way of addressing by bits 1 or 0 for broadcast packets it sends. The default broadcast address setting on new systems is 1.

Like a broadcast address, a multicast address is a single address that represents a group of devices on the network. Unlike a broadcast address, machines using a multicast address must first wish to receive packets directed to this address. The message sent to the broadcast address is received by all machines supporting the IP protocol, regardless of whether they are interested in its content or not. Some routing protocols use multicast addresses as the destination address for routing information that is sent periodically. This allows such messages to be easily ignored by machines that are not interested in updating the routing information. In turn, the broadcast must be received and parsed by all machines, including hosts that do not support the IP protocol. Only after receiving such a packet, the machine can determine whether it is interesting. This is due to the fact that broadcast packets are handled at the hardware level. The result is that this type of packet is sent to all network adapters, regardless of whether they are supported by the IP protocol or another network protocol that does not understand broadcast messages. Hosts working with a different protocol should reject broadcast packets but doing so requires the host to process the packet to confirm that it is not interested in broadcast packets.

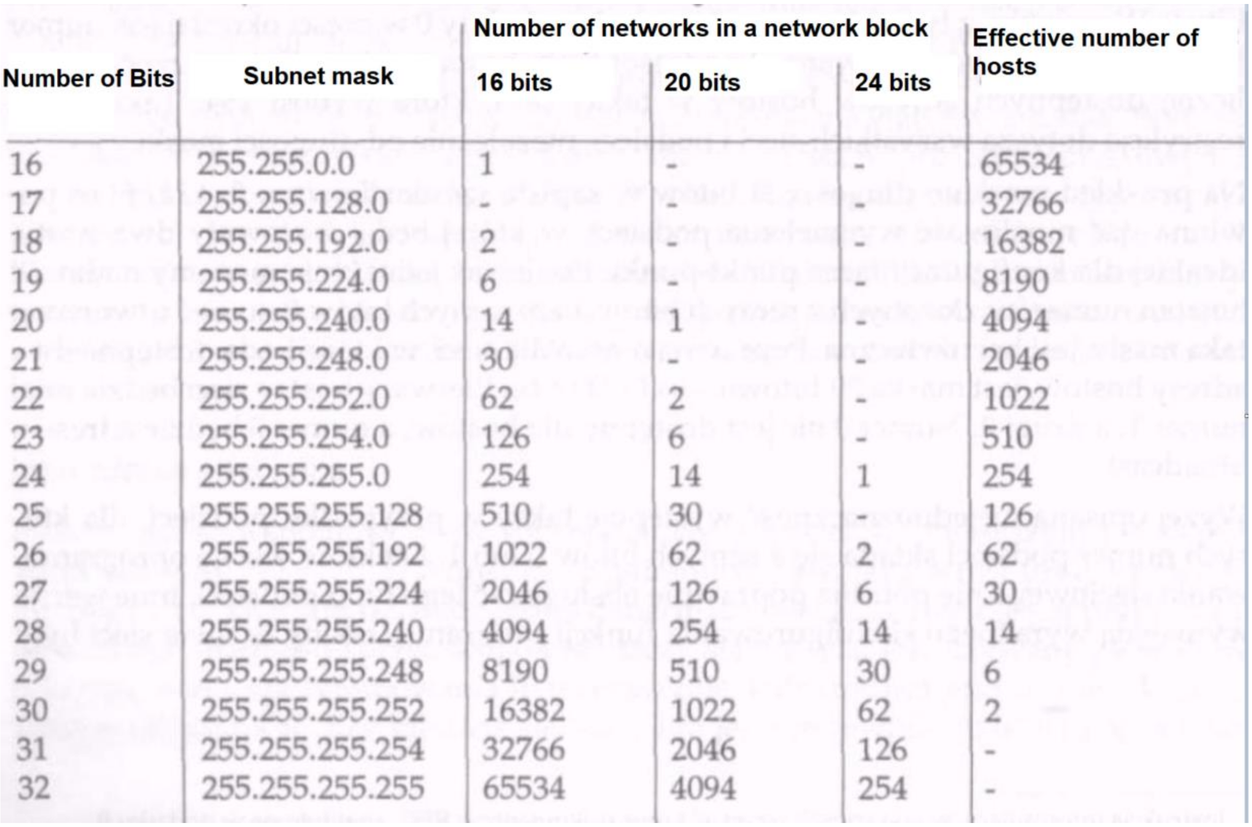
Other special addresses

Two special IP addresses should also be mentioned. The first is the loopback address, 127.0.0.1. This address is defined as the address of the software loopback interface running on the machine. This address is not assigned to any hardware interface and does not connect to the network. It is mainly used to test IP software on a machine that is not connected to the network and regardless of whether the network interface or its drivers are working properly. It can also be used on the local machine as an interface address that is always active and reachable by the software, regardless of the current state of the hardware interfaces. This address can be used, for example for addressing client software references with a roaming server on the same machine, without having to use an external host IP address. The IP protocol specification requires that this address, as well as the entire network 127.0.0.0/8, should never be assigned to an external machine interface. If this happens, these addresses will be dropped by any host or router that receives packets so addressed. Note that this address violates the rule that the IP address uniquely identifies the host, since all hosts on the IP network use the same address for the loopback interface. The second special IP address is 0.0.0.0. In addition to using it as a local network broadcast address in legacy software, some routing protocols treat it as a captured address or default route.

Usable addresses for the given netmask

Until now, we assumed that up to 256 hosts could be placed in each 24-bit mask network. It is not entirely true to recall that the address containing bits 1 in the host number part is the broadcast address. Also, remember that in some older implementations they are used to define the broadcast address 0 bits. Therefore, addresses containing 1 and 0 bits in the host number portion cannot be used to address a host on the network. This gives the actual number of available host addresses on such a network, which is 254. The same restrictions apply to all networks and subnets, regardless of the mask length. For example, a 31 bits long hexadecimal mask 0xFFFFFFFE should allow for a subnetwork with two hosts, ideal for a point-to-point link configuration. However, since we cannot assign numbers to hosts only 1 or 0 bits, the network formed by such a mask is useless. A valid mask for a network with two host addresses is a 30-bit mask - 0xFFFFFFFC. The first host on the network will be l and the second will be 2. Number 0 is not available to hosts and number 3 will be the broadcast address. The above-described ambiguity also occurs in the case of subnets for which the subnetwork number consists of only bits 0 or 1. Some versions of the network software cannot properly handle this type of subnets. Other versions require explicit program features to be configured so that these two networks are handled correctly.

Table 1-2 shows the number of subnets and hosts for all subnet masks in the three different sized network blocks. For example, if your network block is 16 bits long, you can use a 25-bit subnet mask to get 510 subnets and 126 hosts on each. However, if the block length network is 20 bits, the same 25-bit mask can address 30 subnets and 126 hosts in each of them. Note that some masks do not create a useful number of subnets. Such cases are marked with a horizontal line. You can easily break similar network numbers into network blocks of different lengths. As you think about choosing a mask for your subnets, keep in mind the examples in the table below.



Number of Bits	Subnet mask	Number of networks in a network block			Effective number of hosts
		16 bits	20 bits	24 bits	
16	255.255.0.0	1	-	-	65534
17	255.255.128.0	-	-	-	32766
18	255.255.192.0	2	-	-	16382
19	255.255.224.0	6	-	-	8190
20	255.255.240.0	14	1	-	4094
21	255.255.248.0	30	-	-	2046
22	255.255.252.0	62	2	-	1022
23	255.255.254.0	126	6	-	510
24	255.255.255.0	254	14	1	254
25	255.255.255.128	510	30	-	126
26	255.255.255.192	1022	62	2	62
27	255.255.255.224	2046	126	6	30
28	255.255.255.240	4094	254	14	14
29	255.255.255.248	8190	510	30	6
30	255.255.255.252	16382	1022	62	2
31	255.255.255.254	32766	2046	126	-
32	255.255.255.255	65534	4094	254	-

Table 1-2. Number of subnets and hosts depending on the length of the mask and network

IP routing algorithm

In an IP network, each device makes its own routing decisions. The algorithm used to make these decisions is the same, whether it is a host or a router. The sending computer needs to define the entire path through the network to its destination. It only has to point to the next device or next-hop that is part of the full route. Then the packet is sent to the indicated device which is responsible for indicating the next-hop direction towards the destination. This process is repeated until the packet is finally delivered to the device to which it was addressed. Information about subsequent hops towards the destination address is stored in the routing table. Each line in this table describes one IP network, subnet, or host, and the next-hop address that goes there.

Standard IP routing

Although most routers can route packets over classless IP networks, some routers still use a routing algorithm that is associated with the class of network in which the destination address resides. This class-based routing algorithm is as follows:

For the target IP address:

```
if (we have a direct route to the host)
    read the next jump address from the found entry,
    send the packet to the found next-hop address.

if (we do not have a direct route to the host)

    if (we have an interface belonging to this network)
        we define the subnet mask based on information from our interface

    if (we do not have an interface belonging to this network)
        we define the subnet mask based on the address class

    we put the mask on the address to get the subnet address

    if (we have an interface in this subnet)
        we send the package to the addressee.

    if (we don't have an interface in this subnet)
        we search the routing table for an entry about this subnet

            if (we find an entry)
                we send the package to the found next-hop address

            if (we don't find an entry)
                we are looking for the default route in the routing table

                if (we have a default route)
                    we send the packet to the address of the next-hop of the default
                    route

                if (we don't have a default route)
                    we drop the packet with the message "destination unreachable"
```

The algorithm first checks a route that goes directly to the host. A direct route has placed an entry in the routing table that exactly describes the route to the IP address where the packet is destined. Such notation can be used to indicate the device operating on the other side of the serial point-to-point link. If a direct route is not found in the routing table, the algorithm tries to determine the subnet mask for the network destination. For remote networks (those to which the sending computer is not directly attached) the routing table does not contain any information about the subnet mask used, so the natural mask from the network class is used. If there is a direct connection to the network, the mask is determined based on the configuration of the host's network interface. This interface may or may not be attached to the subnet on which it is located the destination address, but the algorithm assumes that the netmask is the same. As a result, class routing will not function properly in a network with different subnet masks in different areas, unless the network is very carefully configured by the administrator to avoid ambiguity. When the algorithm determines a subnet mask for the network to which packets are sent, the destination address is masked with this one mask to get a subnet number that will be used as a key for the routing table lookup. If the algorithm finds that the host is attached directly to this network, the packet is sent directly to the addressee. Otherwise, the routing table is searched for a record with route information to the given subnetwork, and when such a record is found, the next-hop address is determined. If this fails, the algorithm treats it as a last resort when searching for a record with information about default routing. The default routing usually indicates a smarter router (one with a more complete routing table), but it can also point to a router that is closer to the main IP network (core) than the originator. If the algorithm is unable to determine the next transition, it returns a message that the destination address is not reachable. This information is sent directly to the user's program (if the sending computer cannot find the next transition) or using the Internet Control Message Protocol (ICMP).

Classless IP routing

With the introduction of supernets, the routing algorithm must be updated to work with an arbitrarily defined portion of the IP address space. In each entry in the routing table, it is necessary to include the destination address and the next-hop address, as well as a mask that will allow determining the size of the address space described by this entry. Adding this mask to a record in the routing table allows the class routing algorithm to be generalized to a classless algorithm. The implementation of the search part in such an algorithm is much more complicated than in the case of the class algorithm, but the algorithm itself is much simpler:

```
For the target IP address:
Search the routing table for the longest prefix that matches the given address,

        if (matching entry found):
                read the next jump address from the found entry,
                send the packet to the found next-hop address.

        if (no matching entry found):
                drop packet with message "destination unreachable"
```

The prefix here means the remainder of the IP address after masking, e.g .:

192.168.44.1/8 gives the prefix 192.

192.168.44.1/16 gives the prefix 192.168.

When selecting a route, only the destination address prefix is compared with that of the entry in the routing table. See the example at the end of this chapter.

The first visible difference is that this algorithm is much simpler and less detailed than an algorithm based on nets of classes. Placing the netmasks in the routing table allows you to reduce most of the unusual actions necessary to be performed in the class algorithm. For example, the routes to the host are records in this algorithm with the mask 255.255.255.255. Since such 32-bit masks always correspond to destination addresses with a prefix longer than any subnetwork, network, or supernetwork, they are always preferred over less unambiguous routes, similar to the class algorithm. Also, the default route, if any, is saved as a record with the destination address 0.0.0.0 and mask 0.0.0.0. If this mask is used on any destination address, the result will always be 0.0.0.0, which always matches the destination address in this record. The resulting prefix, however, will always be shorter than any other specified route that may lead to that network, subnetwork, or supernetwork, with the result that that route remains the last select route.

A useful consequence of the "longest match" requirement is that a less defined route, e.g. to a supernetwork and a better-defined route to a subnet, can be included in the routing table. Both of these routes lead to the packet destination but have a different next-hop address. This allows one route record to be used to most superlattices and to add route records to fill in the routing gaps resulting from this general record. While this is useful, avoid creating too many holes in the network block or address block as it prevents the creation of small, more efficient routing tables. Remember that if you have holes in a network block or an address block, in addition to the routing table entries that define the route to the supernetwork or network, you must add routes for every hole.

An example of a system routing table with support for classless routing:

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     192.168.44.1    255.255.255.0   UG    0      0        0 eth0
192.168.44.0    *               255.255.255.0   U     0      0        0 eth0
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         *               0.0.0.0         U     0      0        0 ppp0
```

which means:

1. If you have a package to ship to an address that begins with 192.168.1. Then send it to the router with the address 192.168.44.1 which is connected to the network on your eth0 interface.

2. If you have a package to ship to an address that starts with 192.168.44. Then just send it to the eth0 interface - the recipient it's also connected directly to that network, so it will pick it up.

3. If you have a packet to send to an address that starts with 127, send it to the lo (loopback) interface.

4. If you have a packet to send to any address (0-bit mask) send it via the ppp0 interface.

Note that if the address matches several routes, the one with the longest mask is used (e.g.: 192.168.44.0/24 is more important than 192.168.0.0/16)

Maintaining the routing table

Since every device on the IP network sends an IP packet to the next-hop (next-hop - without remembering the entire route of this packet), all the way to the destination, all devices, especially all routers, must constantly create an image of the routes leading in each of the directions. In other words, synchronizing the routing tables between the cooperating routers is most important. To understand why it is necessary, consider the case where Router A and Router B believe that the latter is the correct next-hop route to 10.0.0.1. When Router A receives a packet destined for 10.0.0.1, it will forward it to Router B. Router B, in turn, looks at its routing table and determines that the next-hop router for that address is Router A, then sends the packet back to that Router. The result is a routing loop that more than two routers can create.

Routing table synchronization can be performed in several ways. The easiest way to learn and implement is static routing. In static routing, each router is manually configured, and a list of destination addresses and the next-pass address for those addresses are entered into its table. In this case, the routing table is stored in a configuration file on a persistent medium. It is the job of the network administrator to make sure that all routing tables of the routers that work together are consistent. It is up to the administrator to check for any routing loops and that all directions are reachable from the cooperating routers.

The simplicity of setting up static routing relates to networks with packets going out to a few points, or to end networks that have only one or two connections to the rest of the network. However, this configuration is not without its drawbacks. The most important of these is that static routing cannot adapt the network configuration to failures that occur in it, nor can it take advantage of an alternative route to the destination point. Moreover, as the number of packet forwarding directions, as well as the number of routers, increases, updating the routing tables as the network topology changes become difficult and time-consuming.

More flexible solutions use routing protocols that allow routers to dynamically create routing tables based on information from other routers on the network. Many such protocols have been developed and implemented. Generally speaking, the routers talk to each other using a protocol that can dynamically determine the current network topology. Based on this information, each router determines the routers (one or more) of the next pass to the destination in an attempt to determine the best route. If nothing interferes with communication between the routers, and if all of them use the protocol correctly, they will compute the routing tables that match.

Among the extremely different solutions of static routing and dynamic routing, there are many solutions that combine the advantages of dynamic functions and static functions. Such hybrid routing methods provide a solution that has the advantages of the flexibility of dynamic routing and the simplicity of static routing. For example, network routers may use dynamic routing, and hosts that are attached to single networks may have a default route configured. It is also possible to configure a router to have several static routes in its table, for example, to areas of the network outside the administrator-controlled domain, and to broadcast routes to other routers using a dynamic routing protocol.