Understanding IP Addressing:
Everything You Ever Wanted To Know

Class C
Prefix -Length
Class A
All-1s
CIDR
/24
Network-Prefix
Subnetting
All-0s
Supernetting
Mask
Extended-Network Prefix
Longest Match
Classful
/8
Host-Number
VLSM
/16
Class B
Classless

Chuck Semeria
NSD Marketing
3Com Corporation
April 26, 1996

Introduction

In the mid-1990's, the Internet is a dramatically different network than when it was first established in the early 1980's. Today, the Internet has entered the public consciousness as the world's largest public data network, doubling in size every nine months. This is reflected in the tremendous popularity of the World Wide Web (WWW), the opportunities that businesses see in reaching customers from virtual storefronts, and the emergence of new types and methods of doing business. It is clear that expanding business and social awareness will continue to increase public demand for access to resources on the Internet.

There is a direct relationship between the value of the Internet and the number of sites connected to the Internet. As the Internet grows, the value of each site's connection to the Internet increases because it provides the organization with access to an ever expanding user/customer population.

Internet Scaling Problems

Over the past few years, the Internet has experienced two major scaling issues as it has struggled to provide continuous and uninterrupted growth:

- The eventual exhaustion of the IPv4 address space
- The ability to route traffic between the ever increasing number of networks that comprise the Internet

The first problem is concerned with the eventual depletion of the IP address space. The current version of IP, IP version 4 (IPv4), defines a 32-bit address which means that there are only 2^{32} (4,294,967,296) IPv4 addresses available. This might seem like a large number of addresses, but as new markets open and a significant portion of the world's population becomes candidates for IP addresses, the finite number of IP addresses will eventually be exhausted.

The address shortage problem is aggravated by the fact that portions of the IP address space have not been efficiently allocated. Also, the traditional model of classful addressing does not allow the address space to be used to its maximum potential. The Address Lifetime Expectancy (ALE) Working Group of the IETF has expressed concerns that if the current address allocation policies are not modified, the Internet will experience a near to medium term exhaustion of its unallocated address pool. If the Internet's address supply problem is not solved, new users may be unable to connect to the global Internet!
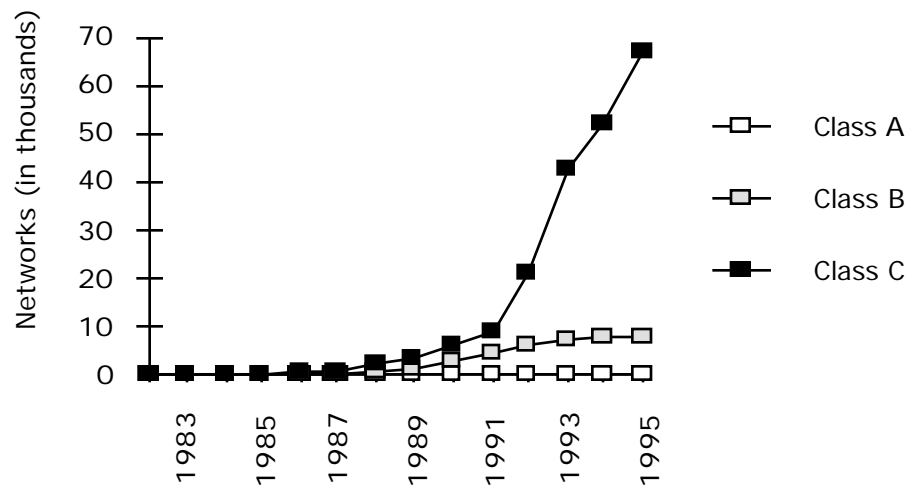
Figure 1: Assigned and Allocated Network Numbers

The second problem is caused by the rapid growth in the size of the Internet routing tables. Internet backbone routers are required to maintain complete routing information for the Internet. Over recent years, routing tables have experienced exponential growth as increasing numbers of organizations connect to the Internet - in December 1990 there were 2,190 routes, in December 1992 there were 8,500 routes, and in December 1995 there were 30,000+ routes.
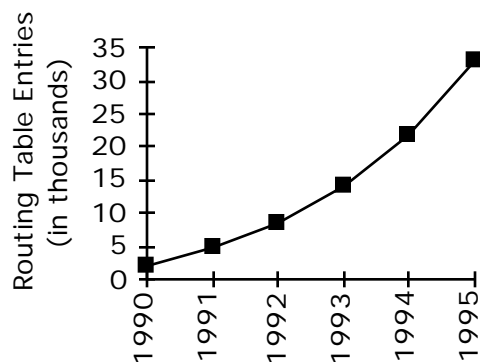

Figure 2: Growth of Internet Routing Tables

Unfortunately, the routing problem cannot be solved by simply installing more router memory and increasing the size of the routing tables. Other factors related to the capacity problem include the growing demand for CPU horsepower to compute routing table/topology changes, the increasingly dynamic nature of WWW connections and their effect on router forwarding caches, and the sheer volume of information that needs to be managed by people and machines. If the number of entries in the global routing table is allowed to increase without bounds, core routers will be forced to drop routes and portions of the Internet will become unreachable!

The long term solution to these problems can be found in the widespread deployment of IP Next Generation (IPng or IPv6) towards the turn of the century. However, while the Internet community waits for IPng, IPv4 will need to be patched and modified so that the Internet can continue to provide the universal connectivity we have come to expect. This patching process may cause a tremendous amount of pain and may alter some of our fundamental concepts about the Internet.

Classful IP Addressing

When IP was first standardized in September 1981, the specification required that each system attached to an IP-based internet be assigned a unique, 32-bit Internet address value. Some systems, such as routers which have interfaces to more than one network, must be assigned a unique IP address for each network interface.

The first part of an Internet address identifies the network on which the host resides, while the second part identifies the particular host on the given network. This created the two-level addressing hierarchy which is illustrated in Figure 3.
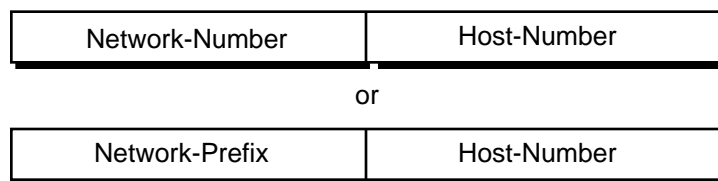
Network-Number	Host-Number

or

Network-Prefix	Host-Number

Figure 3: Two-Level Internet Address Structure

In recent years, the network-number field has been referred to as the "network-prefix" because the leading portion of each IP address identifies the network number. All hosts on a given network share the same network-prefix but must have a unique host-number. Similarly, any two hosts on different networks must have different network-prefixes but may have the same host-number.

Primary Address Classes

In order to provide the flexibility required to support different size networks, the designers decided that the IP address space should be divided into three different address classes - Class A, Class B, and Class C. This is often referred to as "classful" addressing because the address space is split into three predefined classes, groupings, or categories. Each class fixes the boundary between the network-prefix and the host-number at a different point within the 32-bit address. The formats of the fundamental address classes are illustrated in Figure 4.
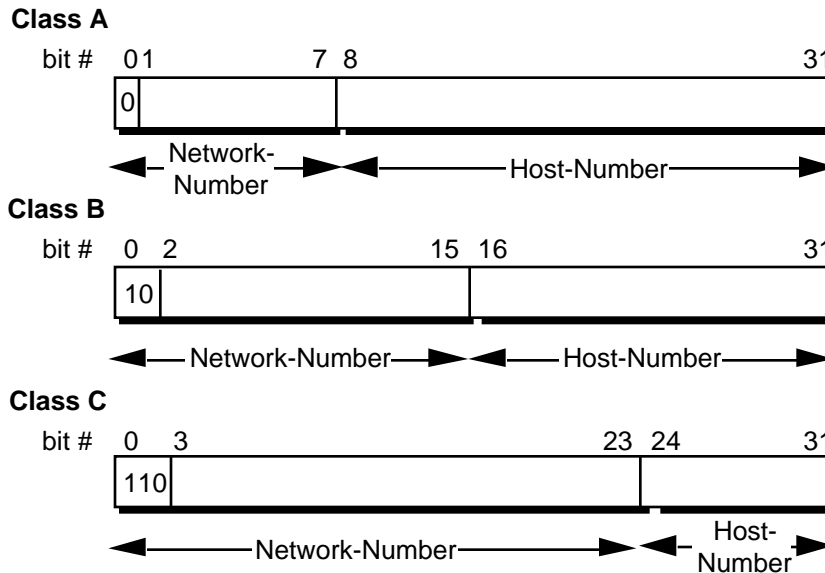
Class A

bit #   0 1   7 8   31

Class B

bit #   0   2   15 16   31

Class C

bit #   0   3   23 24   31

Figure 4: Principle Classful IP Address Formats

One of the fundamental features of classful IP addressing is that each address contains a self-encoding key that identifies the dividing point between the network-prefix and the host-number. For example, if the first two bits of an IP address are 1-0, the dividing point falls between the 15th and 16th bits. This simplified the routing system during the early years of the Internet because the original routing protocols did not supply a "deciphering key" or "mask" with each route to identify the length of the network-prefix.

Class A Networks (/8 Prefixes)

Each Class A network address has an 8-bit network-prefix with the highest order bit set to 0 and a seven-bit network number, followed by a 24-bit host-number. Today, it is no longer considered 'modern' to refer to a Class A network. Class A networks are now referred to as "/8s" (pronounced "slash eight" or just "eights") since they have an 8-bit network-prefix.

A maximum of 126 (2^7-2) /8 networks can be defined. The calculation requires that the 2 is subtracted because the /8 network 0.0.0.0 is reserved for use as the default route and the /8 network 127.0.0.0 (also written 127/8 or 127.0.0.0/8) has been reserved for the "loopback" function. Each /8 supports a maximum of 16,777,214 (2^{24}-2) hosts per network. The host calculation requires that 2 is subtracted because the all-0s ("this network") and all-1s ("broadcast") host-numbers may not be assigned to individual hosts.

Since the /8 address block contains 2^{31} (2,147,483,648) individual addresses and the IPv4 address space contains a maximum of 2^{32} (4,294,967,296) addresses, the /8 address space is 50% of the total IPv4 unicast address space.

Class B Networks (/16 Prefixes)

Each Class B network address has a 16-bit network-prefix with the two highest order bits set to 1-0 and a 14-bit network number, followed by a 16-bit host-number. Class B networks are now referred to as"/16s" since they have a 16-bit network-prefix.

A maximum of 16,384 (2^{14}) /16 networks can be defined with up to 65,534 (2^{16}-2) hosts per network. Since the entire /16 address block contains 2^{30} (1,073,741,824) addresses, it represents 25% of the total IPv4 unicast address space.

Class C Networks (/24 Prefixes)

Each Class C network address has a 24-bit network-prefix with the three highest order bits set to 1-1-0 and a 21-bit network number, followed by an 8-bit host-number. Class C networks are now referred to as "/24s" since they have a 24-bit network-prefix.

A maximum of 2,097,152 (2^{21}) /24 networks can be defined with up to 254 (2^{8}-2) hosts per network. Since the entire /24 address block contains 2^{29} (536,870,912) addresses, it represents 12.5% (or 1/8th) of the total IPv4 unicast address space.

Other Classes

In addition to the three most popular classes, there are two additional classes. Class D addresses have their leading four-bits set to 1-1-1-0 and are used to support IP Multicasting. Class E addresses have their leading four-bits set to 1-1-1-1 and are reserved for experimental use.

Dotted-Decimal Notation

To make Internet addresses easier for human users to read and write, IP addresses are often expressed as four decimal numbers, each separated by a dot. This format is called "dotted-decimal notation."

Dotted-decimal notation divides the 32-bit Internet address into four 8-bit (byte) fields and specifies the value of each field independently as a decimal number with the fields separated by dots. Figure 5 shows how a typical /16 (Class B) Internet address can be expressed in dotted decimal notation.
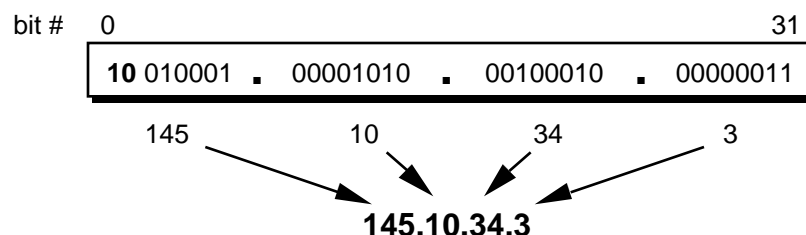

Figure 5: Dotted-Decimal Notation

Table 1 displays the range of dotted-decimal values that can be assigned to each of the three principle address classes. The "xxx" represents the host-number field of the address which is assigned by the local network administrator.

Table 1: Dotted-Decimal Ranges for Each Address Class

Address Class	Dotted-Decimal Notation Ranges
A (/8 prefixes)	1.xxx.xxx.xxx through 126.xxx.xxx.xxx
B (/16 prefixes)	128.0.xxx.xxx through 191.255.xxx.xxx
C (/24 prefixes)	192.0.0.xxx through 223.255.255.xxx

Unforeseen Limitations to Classful Addressing

The original designers never envisioned that the Internet would grow into what it has become today. Many of the problems that the Internet is facing today can be traced back to the early decisions that were made during its formative years.

- During the early days of the Internet, the seemingly unlimited address space allowed IP addresses to be allocated to an organization based on its request rather than its actual need. As a result, addresses were freely assigned to those who asked for them without concerns about the eventual depletion of the IP address space.

- The decision to standardize on a 32-bit address space meant that there were only 2^{32} (4,294,967,296) IPv4 addresses available. A decision to support a slightly larger address space would have exponentially increased the number of addresses thus eliminating the current address shortage problem.

- The classful A, B, and C octet boundaries were easy to understand and implement, but they did not foster the efficient allocation of a finite address space. Problems resulted from the lack of a network class that was designed to support medium-sized organizations. A /24, which supports 254 hosts, is too small while a /16, which supports 65,534 hosts, is too large. In the past, the Internet has assigned sites with several hundred hosts a single /16 address instead of a couple of /24s addresses. Unfortunately, this has resulted in a premature depletion of the /16 network address space. The only readily available addresses for medium-size organizations are /24s which have the potentially negative impact of increasing the size of the global Internet's routing table.

The subsequent history of Internet addressing is focused on a series of steps that overcome these addressing issues and have supported the growth of the global Internet.

Additional Practice with Classful Addressing

Please turn to Appendix B for practical exercises to further your understanding of Classful IP Addressing.
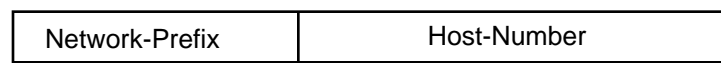
Subnetting

In 1985, RFC 950 defined a standard procedure to support the subnetting, or division, of a single Class A, B, or C network number into smaller pieces. Subnetting was introduced to overcome some of the problems that parts of the Internet were beginning to experience with the classful two-level addressing hierarchy:

- Internet routing tables were beginning to grow.

- Local administrators had to request another network number from the Internet before a new network could be installed at their site.

Both of these problems were attacked by adding another level of hierarchy to the IP addressing structure. Instead of the classful two-level hierarchy, subnetting supports a three-level hierarchy. Figure 6 illustrates the basic idea of subnetting which is to divide the standard classful host-number field into two parts - the subnet-number and the host-number on that subnet.

Two-Level Classful Hierarchy

Network-Prefix	Host-Number

Three-Level Subnet Hierarchy

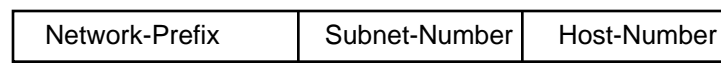| Network-Prefix	Subnet-Number	Host-Number

Figure 6: Subnet Address Hierarchy

Subnetting attacked the expanding routing table problem by ensuring that the subnet structure of a network is never visible outside of the organization's private network. The route from the Internet to any subnet of a given IP address is the same, no matter which subnet the destination host is on. This is because all subnets of a given network number use the same network-prefix but different subnet numbers. The routers within the private organization need to differentiate between the individual subnets, but as far as the Internet routers are concerned, all of the subnets in the organization are collected into a single routing table entry. This allows the local administrator to introduce arbitrary complexity into the private network without affecting the size of the Internet's routing tables.

Subnetting overcame the registered number issue by assigning each organization one (or at most a few) network number(s) from the IPv4 address space. The organization was then free to assign a distinct subnetwork number for each of its internal networks. This allows the organization to deploy additional subnets without needing to obtain a new network number from the Internet.
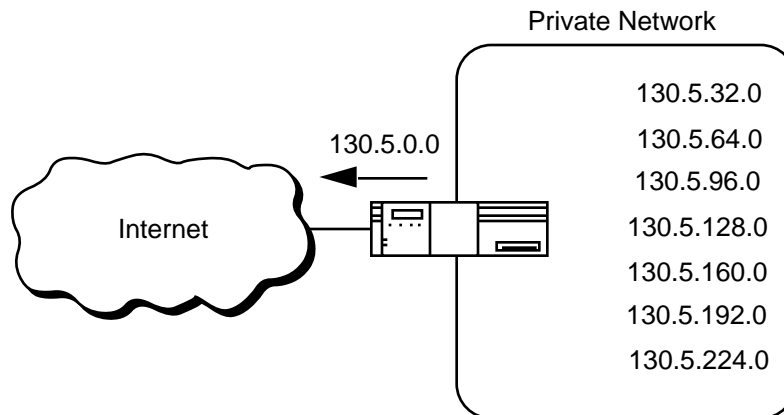
Private Network

130.5.0.0

130.5.32.0
130.5.64.0
130.5.96.0
130.5.128.0
130.5.160.0
130.5.192.0
130.5.224.0

Internet

Figure 7: Subnetting Reduces the Routing Requirements of the Internet

In Figure 7, a site with several logical networks uses subnet addressing to cover them with a single /16 (Class B) network address. The router accepts all traffic from the Internet addressed to network 130.5.0.0, and forwards traffic to the interior subnetworks based on the third octet of the classful address. The deployment of subnetting within the private network provides several benefits:

-   The size of the global Internet routing table does not grow because the site administrator does not need to obtain additional address space and the routing advertisements for all of the subnets are combined into a single routing table entry.

-   The local administrator has the flexibility to deploy additional subnets without obtaining a new network number from the Internet.

-   Route flapping (i.e., the rapid changing of routes) within the private network does not affect the Internet routing table since Internet routers do not know about the reachability of the individual subnets - they just know about the reachability of the parent network number.

Extended-Network-Prefix

Internet routers use only the network-prefix of the destination address to route traffic to a subnetted environment. Routers within the subnetted environment use the extended-network-prefix to route traffic between the individual subnets. The extended-network-prefix is composed of the classful network-prefix and the subnet-number.
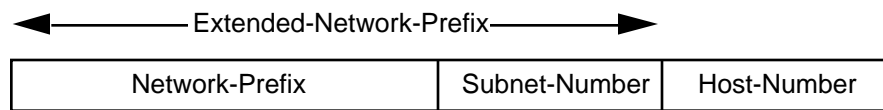
←————— Extended-Network-Prefix —————→

Network-Prefix	Subnet-Number	Host-Number

Figure 8: Extended-Network-Prefix

The extended-network-prefix has traditionally been identified by the subnet mask. For example, if you have the /16 address of 130.5.0.0 and you want to use the entire third octet to represent the subnet-number, you need to specify a subnet mask of 255.255.255.0. The bits in the subnet mask and the Internet address have a one-to-one correspondence. The bits of the subnet mask are set to 1 if the system examining the address should treat the corresponding bit in the IP address as part of the extended-network-prefix. The bits in the mask are set to 0 if the system should treat the bit as part of the host-number. This is illustrated if Figure 9.
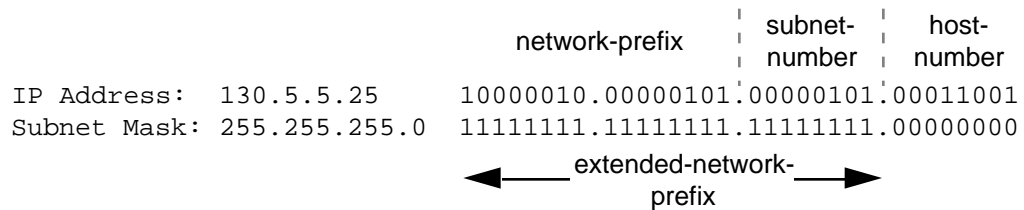
```
                                                  subnet-        host-
                                 network-prefix   number         number

IP Address:   130.5.5.25         10000010.00000101.00000101.00011001
Subnet Mask:  255.255.255.0      11111111.11111111.11111111.00000000
                                      ◄────── extended-network-──────►
                                                  prefix
```

Figure 9: Subnet Mask

The standards describing modern routing protocols often refer to the extended-network-prefix-length rather than the subnet mask. The prefix length is equal to the number of contiguous one-bits in the traditional subnet mask. This means that specifying the network address 130.5.5.25 with a subnet mask of 255.255.255.0 can also be expressed as 130.5.5.25/24. The /<prefix-length> notation is more compact and easier to understand than writing out the mask in its traditional dotted-decimal format. This is illustrated in Figure 10.
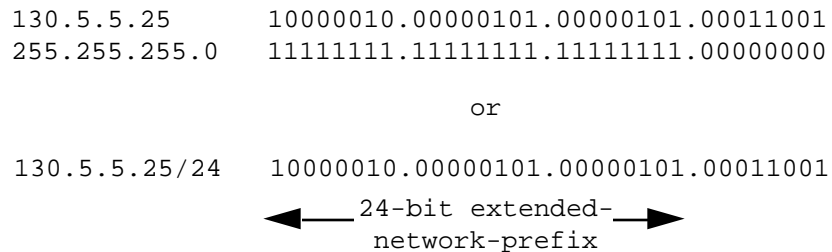
```
130.5.5.25         10000010.00000101.00000101.00011001
255.255.255.0      11111111.11111111.11111111.00000000


                            or


130.5.5.25/24      10000010.00000101.00000101.00011001
                       ◄───── 24-bit extended-──────►
                               network-prefix
```

Figure 10: Extended-Network-Prefix Length

However, it is important to note that modern routing protocols still carry the subnet mask. There are no Internet standard routing protocols that have a one-byte field in their header that contains the number of bits in the extended-network prefix. Rather, each routing protocol is still required to carry the complete four-octet subnet mask.

Subnet Design Considerations

The deployment of an addressing plan requires careful thought on the part of the network administrator. There are four key questions that must be answered before any design should be undertaken:

1) How many total subnets does the organization need today?

2) How many total subnets will the organization need in the future?

3) How many hosts are there on the organization's largest subnet today?

4) How many hosts will there be on the organization's largest subnet in the future?

The first step in the planning process is to take the maximum number of subnets required and round up to the nearest power of two. For example, if a organization needs 9 subnets, 2^3 (or 8) will not provide enough subnet addressing space, so the network administrator will need to round up to 2^4 (or 16). When performing this assessment, it is critical that the network administrator always allow adequate room for future growth. For example, if 14 subnets are required today, then 16 subnets might not be enough in two years when the 17th subnet needs to be deployed. In this case, it might be wise to allow for more growth and select 2^5 (or 32) as the maximum number of subnets.

The second step is to make sure that there are enough host addresses for the organization's largest subnet. If the largest subnet needs to support 50 host addresses today, 2^5 (or 32) will not provide enough host address space so the network administrator will need to round up to 2^6 (or 64).

The final step is to make sure that the organization's address allocation provides enough bits to deploy the required subnet addressing plan. For example, if the organization has a single /16, it could easily deploy 4-bits for the subnet-number and 6-bits for the host number. However, if the organization has several /24s and it needs to deploy 9 subnets, it may be required to subnet each of its /24s into four subnets (using 2 bits) and then build the internet by combining the subnets of 3 different /24 network numbers. An alternative solution, would be to deploy network numbers from the private address space (RFC 1918) for internal connectivity and use a Network Address Translator (NAT) to provide external Internet access.

Subnet Example #1

Given

An organization has been assigned the network number 193.1.1.0/24 and it needs to define six subnets. The largest subnet is required to support 25 hosts.

Defining the Subnet Mask / Extended-Prefix Length

The first step is to determine the number of bits required to define the six subnets. Since a network address can only be subnetted along binary boundaries, subnets must be created in blocks of powers of two [2 (2^1), 4 (2^2), 8 (2^3), 16 (2^4), etc.]. Thus, it is impossible to define an IP address block such that it contains exactly six subnets. For this example, the network administrator must define a block of 8 (2^3) and have two unused subnets that can be reserved for future growth.

Since 8 = 2^3, three bits are required to enumerate the eight subnets in the block. In this example, the organization is subnetting a /24 so it will need three more bits, or a /27, as the extended-network-prefix. A 27-bit extended-network-prefix can be expressed in dotted-decimal notation as 255.255.255.224. This is illustrated in Figure 11.
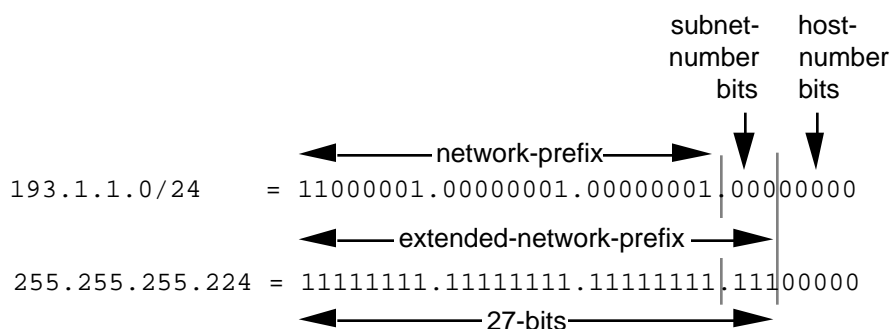
```
                                          subnet-   host-
                                          number    number
                                          bits      bits
                                            ↓         ↓
                      ←————network-prefix————→
193.1.1.0/24      = 11000001.00000001.00000001.000 00000
                        ←———— extended-network-prefix ———→
255.255.255.224  = 11111111.11111111.11111111.111 00000
                      ←——————————27-bits——————————→
```

Figure 11: Example #1 - Defining the Subnet Mask/Extended-Prefix Length

A 27-bit extended-network-prefix leaves 5 bits to define host addresses on each subnet. This means that each subnetwork with a 27-bit prefix represents a contiguous block of 2^5 (32) individual IP addresses. However, since the all-0s and all-1s host addresses cannot be allocated, there are 30 (2^5 -2) assignable host addresses on each subnet.

Defining Each of the Subnet Numbers

The eight subnets will be numbered 0 through 7. Throughout the remainder of this paper, the XXX_2 notation indicates the binary representation of the number. The 3-bit binary representation of the decimal values 0 through 7 are: 0 (000_2), 1 (001_2), 2 (010_2), 3 (011_2), 4 (100_2), 5 (101_2), 6 (110_2), and 7 (111_2).

In general, to define Subnet #n, the network administrator places the binary representation of n into the bits of the subnet-number field. For example, to define Subnet #6, the network administrator simply places the binary representation of 6 (110_2) into the 3-bits of the subnet-number field.

The eight subnet numbers for this example are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 3-bits representing the subnet-number field:

```
Base Net:   11000001.00000001.00000001.00000000 = 193.1.1.0/24

Subnet #0:  11000001.00000001.00000001.00000000 = 193.1.1.0/27
Subnet #1:  11000001.00000001.00000001.00100000 = 193.1.1.32/27
Subnet #2:  11000001.00000001.00000001.01000000 = 193.1.1.64/27
Subnet #3:  11000001.00000001.00000001.01100000 = 193.1.1.96/27
Subnet #4:  11000001.00000001.00000001.10000000 = 193.1.1.128/27
Subnet #5:  11000001.00000001.00000001.10100000 = 193.1.1.160/27
Subnet #6:  11000001.00000001.00000001.11000000 = 193.1.1.192/27
Subnet #7:  11000001.00000001.00000001.11100000 = 193.1.1.224/27
```

An easy way to check if the subnets are correct is to ensure that they are all multiples of the Subnet #1 address. In this case, all subnets are multiples of 32: 0, 32, 64, 96, ...

The All-0s Subnet and The All-1s Subnet

When subnetting was first defined in RFC 950, it prohibited the use of the all-0s and the all-1s subnet. The reason for this restriction was to eliminate situations that could potentially confuse a classful router. Note that today a router can be both classless and classful at the same time - it could be running RIP-1 (a classful protocol) and BGP-4 (a classless protocol) at the same time.

With respect to the all-0s subnet, a router requires that each routing table update include the route/<prefix-length> pair to differentiate between a route to the all-0s subnet and a route to the entire network. For example, when using RIP-1 which does not supply a mask or prefix-length with each route, the routing advertisements for subnet 193.1.1.0/27 and for network 193.1.1.0/24 are identical - 193.1.1.0. Without somehow knowing the prefix-length or mask, a router cannot tell the difference between a route to the all-0s subnet and the route to the entire network! This is illustrated in Figure 12.
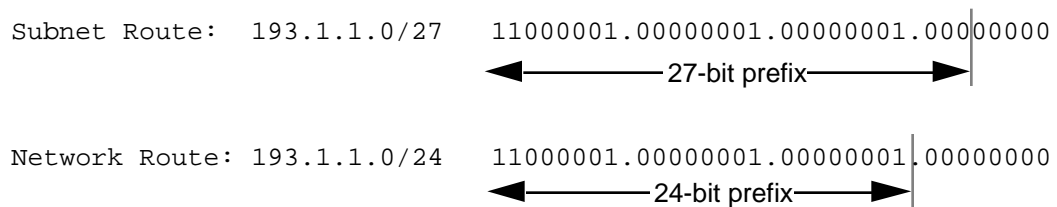
```
Subnet Route:  193.1.1.0/27   11000001.00000001.00000001.000 00000
                                         27-bit prefix

Network Route: 193.1.1.0/24   11000001.00000001.00000001.00000000
                                         24-bit prefix
```

Figure 12: Differentiating Between a Route to the All-0s Subnet and the Entire Network

Regarding the all-1s subnet, a router requires that each routing table entry include the prefix-length so that it can determine if a broadcast (directed or all-subnets) should be sent only to the all-1s subnet or to the entire network. For example, when the routing table does not contain a mask or prefix-length for each route, confusion can occur because the same broadcast address (193.1.1.255) is used for both for the entire network 193.1.1.0/24 and the all-1s subnet 193.1.1.224/27. This is illustrated in Figure 13.
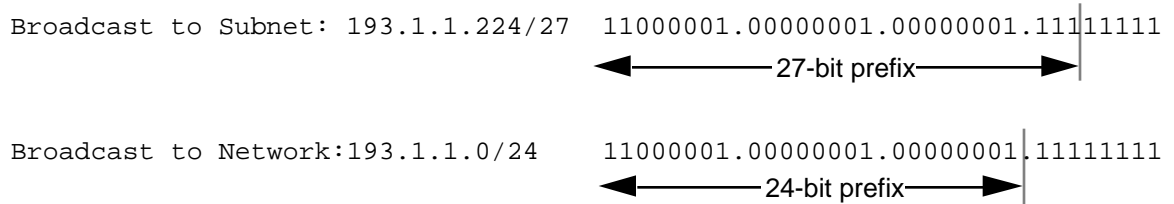
```
Broadcast to Subnet: 193.1.1.224/27    11000001.00000001.00000001.111|11111
                                          ◄───────── 27-bit prefix ─────────►|


Broadcast to Network:193.1.1.0/24      11000001.00000001.00000001|.11111111
                                          ◄───────── 24-bit prefix ─────────►|
```

Figure 13: Identifying a Broadcast to the All-1s Subnet and the Entire Network

Note that by default, NETBuilder software permits the forwarding of traffic to a directed broadcast address but does not forward traffic to the all-subnets broadcast address. The network administrator can modify this behavior via the -IP CONTrol parameter switches FwdSubnetBcast | NoFwdSubnetBcast and FwdAllSubnetBcast | NoFwdAllSubnetBcast.

With the development of routing protocols that supply the mask or prefix-length with each route, the address space defined by the all-0s and all-1s subnets is once again usable despite the cautions in RFC 950. As a result, vendors have begun to accommodate user demand and permit the configuration of the all-0s and all-1s subnets on router interfaces. There are three factors that determine when these subnets can be used with NETBuilder software.

- The interior gateway protocol (IGP)
- The version number of the NETBuilder software release
- The capabilities of other routers in the organization's network.

To support the deployment of the all-0s and all-1s subnets, the IGP must either carry extended-network-prefixes or have a mechanism to map each route to its extended-network-prefix. Both OSPF and I-IS-IS carry extended-network-prefixes, so they support the deployment of the all-0s and all-1s subnets in arbitrarily complex topologies. RIP-1 does not carry extended-network-prefixes but the RcvSubnetMask parameter along with the -RIPIP CONTrol (..[Aggregate| NoAggregate], [DeAggregate| NoDeAggregate]) switches support the deployment of the all-0s and all-1s subnets in *simple* topologies.

NETBuilder software has taken a phased approach in its support for the deployment of the all-0s and all-1s subnets. Table 2 shows the NETBuilder software version number that first implemented the various features that support the deployment of these subnets.

Table 2: NETBuilder Software Features Supporting the All-0s and All-1s Subnets

Release	Feature Supported
6.2	Permits a router interface to be configured with all-0s in the subnet field
7.0.0.6	Correctly learns and forwards to routes with all-0s in the subnet field
7.1	Correctly learns and forwards to routes with all-1s in the subnet field
8.3.0.2 7.2.1	Permits a router interface to be configured with all-1s in the subnet field

Finally, the other routers in the organization's network need to be able to correctly interpret, learn, and forward traffic to other subnetworks with all-0s and all-1s in their subnet number field.

Defining Host Addresses for Each Subnet

According to Internet practices, the host-number field of an IP address cannot contain all 0-bits or all 1-bits. The all-0s host-number identifies the base network (or subnetwork) number, while the all-1s host-number represents the broadcast address for the network (or subnetwork).

In our current example, there are 5 bits in the host-number field of each subnet address. This means that each subnet represents a block of 30 host addresses ($2^5-2 = 30$, note that the 2 is subtracted because the all-0s and the all-1s host addresses cannot be used). The hosts on each subnet are numbered 1 through 30.

In general, to define the address assigned to Host #n of a particular subnet, the network administrator places the binary representation of n into the subnet's host-number field. For example, to define the address assigned to Host #15 on Subnet #2, the network administrator simply places the binary representation of 15 (01111_2) into the 5-bits of Subnet #2's host-number field.

The valid host addresses for Subnet #2 in our example are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 5-bit host-number field:

```
Subnet #2: 11000001.00000001.00000001.01000000 = 193.1.1.64/27

Host #1:   11000001.00000001.00000001.01000001 = 193.1.1.65/27
Host #2:   11000001.00000001.00000001.01000010 = 193.1.1.66/27
Host #3:   11000001.00000001.00000001.01000011 = 193.1.1.67/27
Host #4:   11000001.00000001.00000001.01000100 = 193.1.1.68/27
Host #5:   11000001.00000001.00000001.01000101 = 193.1.1.69/27
              .
              .
```

```
Host #15:   11000001.00000001.00000001.01001111 = 193.1.1.79/27
Host #16:   11000001.00000001.00000001.01010000 = 193.1.1.80/27
              .
              .
              .
Host #27:   11000001.00000001.00000001.01011011 = 193.1.1.91/27
Host #28:   11000001.00000001.00000001.01011100 = 193.1.1.92/27
Host #29:   11000001.00000001.00000001.01011101 = 193.1.1.93/27
Host #30:   11000001.00000001.00000001.01011110 = 193.1.1.94/27
```

The valid host addresses for Subnet #6 are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 5-bit host-number field:

```
Subnet #6:  11000001.00000001.00000001.11000000 = 193.1.1.192/27

Host #1:    11000001.00000001.00000001.11000001 = 193.1.1.193/27
Host #2:    11000001.00000001.00000001.11000010 = 193.1.1.194/27
Host #3:    11000001.00000001.00000001.11000011 = 193.1.1.195/27
Host #4:    11000001.00000001.00000001.11000100 = 193.1.1.196/27
Host #5:    11000001.00000001.00000001.11000101 = 193.1.1.197/27
              .
              .
              .
Host #15:   11000001.00000001.00000001.11001111 = 193.1.1.207/27
Host #16:   11000001.00000001.00000001.11010000 = 193.1.1.208/27
              .
              .
              .
Host #27:   11000001.00000001.00000001.11011011 = 193.1.1.219/27
Host #28:   11000001.00000001.00000001.11011100 = 193.1.1.220/27
Host #29:   11000001.00000001.00000001.11011101 = 193.1.1.221/27
Host #30:   11000001.00000001.00000001.11011110 = 193.1.1.222/27
```

Defining the Broadcast Address for Each Subnet

The broadcast address for Subnet #2 is the all 1's host address or:

```
    11000001.00000001.00000001.01011111 = 193.1.1.95
```

Note that the broadcast address for Subnet #2 is exactly one less than the base address for Subnet #3 (193.1.1.96). This is always the case - the broadcast address for Subnet #n is one less than the base address for Subnet #(n+1).

The broadcast address for Subnet #6 is simply the all 1's host address or:

```
    11000001.00000001.00000001.11011111 = 193.1.1.223
```

Again, the broadcast address for Subnet #6 is exactly one less than the base address for Subnet #7 (193.1.1.224).

Subnet Example #2

Given

An organization has been assigned the network number 140.25.0.0/16 and it needs to create a set of subnets that supports up to 60 hosts on each subnet.

Defining the Subnet Mask / Extended-Prefix Length

The first step is to determine the number of bits required to define 60 hosts on each subnet. Since a block of host address can only be assigned along binary boundaries, host address blocks can only be created in powers of two. This means that it is impossible to create a block that contains exactly 60 host addresses. To support 60 hosts, the network administrator must define a minimum address block of 62 (2^6-2) host addresses. However, this choice would only provide two unused host addresses on each subnet for future growth. Since this does not appear to be adequate to support additional growth, the network administrator elects to define a block of 126 (2^7-2) host addresses and has 66 addresses on each subnet for future growth. A block of 126 host addresses requires 7-bits in the host-number field.

The next step is to determine the subnet mask/extended-prefix length. Since 7-bits of the 32-bit IP address are required for the host-number field, the extended-prefix must be a /25 (25 = 32-7). A 25-bit extended-network-prefix can be expressed in dotted-decimal notation as 255.255.255.128. This is illustrated in Figure 14.
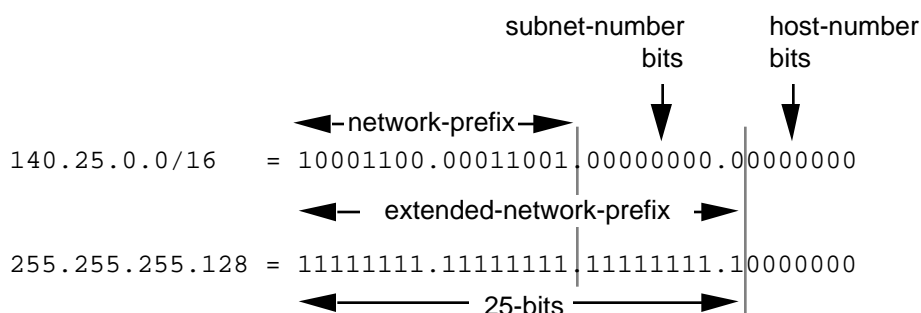
```
                                    subnet-number        host-number
                                         bits               bits
                                          |                  |
                                          v                  v
                  ◄–network-prefix►
140.25.0.0/16   = 10001100.00011001.00000000.00000000
                         ◄— extended-network-prefix ►

255.255.255.128 = 11111111.11111111.11111111.10000000
                   ◄——————————— 25-bits ———————————►
```

Figure 14: Example #2 - Defining the Subnet Mask/Extended-Prefix Length

Figure 14 shows that the 25-bit extended-prefix assigns 9-bits to the subnet number field. Since 2^9 = 512, nine bits allow the definition of 512 subnets. Depending on the organization's requirements, the network administrator could have elected to assign additional bits to the host-number field (allowing more hosts on each subnet) and reduce the number of bits in the subnet-number field (decreasing the total number of subnets that can be defined).

Although this example creates a rather large number of subnets, it provides an interesting example because it illustrates what happens to the dotted-decimal representation of a subnet address when the subnet-number bits extend across an octet boundary. It should be mentioned that the same type of confusion can also occur when the host-number bits extend across an octet boundary.

Defining Each of the Subnet Numbers

The 512 subnets will be numbered 0 through 511. The 9-bit binary representation of the decimal values 0 through 511 are: 0 (000000000_2), 1 (000000001_2), 2 (000000010_2), 3 (000000011_2), ..., 511 (111111111_2). To define subnet #3, the network administrator places the binary representation of 3 (000000011_2) into the 9-bits of the subnet-number field.

The 512 subnet numbers for this example are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 9-bits representing the subnet-number field:

```
Base Net:    10001100.00011001.00000000.00000000 = 140.25.0.0/16

Subnet #0:   10001100.00011001.00000000.00000000 = 140.25.0.0/25
Subnet #1:   10001100.00011001.00000000.10000000 = 140.25.0.128/25
Subnet #2:   10001100.00011001.00000001.00000000 = 140.25.1.0/25
Subnet #3:   10001100.00011001.00000001.10000000 = 140.25.1.128/25
Subnet #4:   10001100.00011001.00000010.00000000 = 140.25.2.0/25
Subnet #5:   10001100.00011001.00000010.10000000 = 140.25.2.128/25
Subnet #6:   10001100.00011001.00000011.00000000 = 140.25.3.0/25
Subnet #7:   10001100.00011001.00000011.10000000 = 140.25.3.128/25
Subnet #8:   10001100.00011001.00000100.00000000 = 140.25.4.0/25
Subnet #9:   10001100.00011001.00000100.10000000 = 140.25.4.128/25
       .
       .
Subnet #510:10001100.00011001.11111111.00000000 = 140.25.255.0/25
Subnet #511:10001100.00011001.11111111.10000000 = 140.25.255.128/25
```

Notice how sequential subnet numbers do not appear to be sequential when expressed in dotted-decimal notation. This can cause a great deal of misunderstanding and confusion since everyone believes that dotted-decimal notation makes it much easier for human users to understand IP addressing. In this example, the dotted-decimal notation obscures rather than clarifies the subnet numbering scheme!

Defining Hosts Addresses for Each Subnet

In this example there are 7 bits in the host-number field of each subnet address. As discussed earlier, this means that each subnet represents a block of 126 host addresses. The hosts on each subnet will be numbered 1 through 126.

The valid host addresses for Subnet #3 are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 7-bit host-number field:

```
Subnet #3:  10001100.00011001.00000001.10000000 = 140.25.1.128/25

Host #1:    10001100.00011001.00000001.10000001 = 140.25.1.129/25
Host #2:    10001100.00011001.00000001.10000010 = 140.25.1.130/25
Host #3:    10001100.00011001.00000001.10000011 = 140.25.1.131/25
Host #4:    10001100.00011001.00000001.10000100 = 140.25.1.132/25
Host #5:    10001100.00011001.00000001.10000101 = 140.25.1.133/25
Host #6:    10001100.00011001.00000001.10000110 = 140.25.1.134/25
               .
               .
               .
Host #62:   10001100.00011001.00000001.10111110 = 140.25.1.190/25
Host #63:   10001100.00011001.00000001.10111111 = 140.25.1.191/25
Host #64:   10001100.00011001.00000001.11000000 = 140.25.1.192/25
Host #65:   10001100.00011001.00000001.11000001 = 140.25.1.193/25
               .
               .
Host #123:  10001100.00011001.00000001.11111011 = 140.25.1.251/25
Host #124:  10001100.00011001.00000001.11111100 = 140.25.1.252/25
Host #125:  10001100.00011001.00000001.11111101 = 140.25.1.253/25
Host #126:  10001100.00011001.00000001.11111110 = 140.25.1.254/25
```

Defining the Broadcast Address for Each Subnet

The broadcast address for Subnet #3 is the all 1's host address or:

```
10001100.00011001.00000001.11111111 = 140.25.1.255
```

As is true in general, the broadcast address for Subnet #3 is exactly one less than the base address for Subnet #4 (140.25.2.0).

Additional Practice with Subnetworks

Please turn to Appendix C for practice exercises to further your understanding of subnetting.

Variable Length Subnet Masks (VLSM)

In 1987, RFC 1009 specified how a subnetted network could use more than one subnet mask. When an IP network is assigned more than one subnet mask, it is considered a network with "variable length subnet masks" since the extended-network-prefixes have different lengths.

RIP-1 Permits Only a Single Subnet Mask

When using RIP-1, subnet masks have to be uniform across the entire network-prefix. RIP-1 allows only a single subnet mask to be used within each network number because it does not provide subnet mask information as part of its routing table update messages. In the absence of this information, RIP-1 is forced to make very simple assumptions about the mask that should be applied to any of its learned routes.

How does a RIP-1 based router know what mask to apply to a route when it learns a new route from a neighbor? If the router has a subnet of the same network number assigned to a local interface, it assumes that the learned subnetwork was defined using the same mask as the locally configured interface. However, if the router does not have a subnet of the learned network number assigned to a local interface, the router has to assume that the network is not subnetted and applies the route's natural classful mask.

Assuming that Port 1 of a router has been assigned the IP address 130.24.13.1/24 and that Port 2 has been assigned the IP address 200.14.13.2/24. If the router learns about network 130.24.36.0 from a neighbor, it applies a /24 mask since Port 1 is configured with another subnet of the 130.24.0.0 network. However, when the router learns about network 131.25.0.0 from a neighbor, it assumes a "natural" /16 mask since it has no other masking information available.

How does a RIP-1 based router know if it should include the subnet-number bits in a routing table update to a RIP-1 neighbor? A router executing RIP-1 will only advertise the subnet-number bits on another port if the update port is configured with a subnet of the same network number. If the update port is configured with a different subnet or network number, the router will only advertise the network portion of the subnet route and "zero-out" the subnet-number field.

For example, assume that Port 1 of a router has been assigned the IP address 130.24.13.1/24 and that Port 2 has been assigned the IP address 200.14.13.2/24. Also, assume that the router has learned about network 130.24.36.0 from a neighbor. Since Port 1 is configured with another subnet of the 130.24.0.0 network, the router assumes that network 130.24.36.0 has a /24 subnet mask. When it comes to advertise this route, it advertises 130.24.36.0 on Port 1, but it only advertises 130.24.0.0 on Port 2.

For these reasons, RIP-1 is limited to only a single subnet mask for each network number. However, there are several advantages to be gained if more than one subnet mask can be assigned to a given IP network number:

- Multiple subnet masks permit more efficient use of an organization's assigned IP address space.

- Multiple subnet masks permit route aggregation which can significantly reduce the amount of routing information at the "backbone" level within an organization's routing domain.

Efficient Use of the Organization's Assigned IP Address Space

VLSM supports more efficient use of an organization's assigned IP address space. One of the major problems with the earlier limitation of supporting only a single subnet mask across a given network-prefix was that once the mask was selected, it locked the organization into a fixed-number of fixed-sized subnets. For example, assume that a network administrator decided to configure the 130.5.0.0/16 network with a /22 extended-network-prefix.
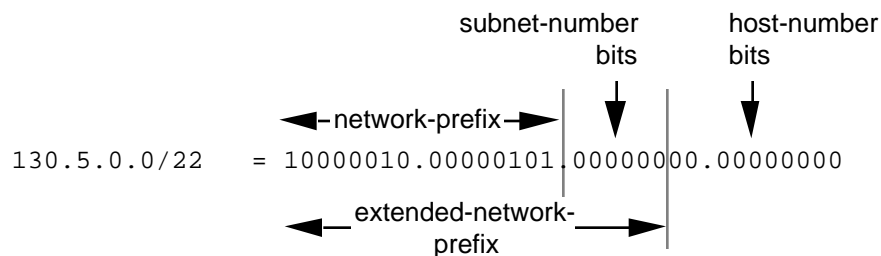


```
                              subnet-number        host-number
                                   bits                bits


                     ◄─network-prefix─►│   ▼           ▼
130.5.0.0/22    = 10000010.00000101.000000 00.00000000
                     ◄─────extended-network-─────►│
                              prefix
```

Figure 15: 130.5.0.0/16 with a /22 Extended-Network Prefix

Please refer to Figure 15. A /16 network with a /22 extended-network prefix permits 64 subnets (2^6), each of which supports a maximum of 1,022 hosts (2^{10}-2). This is fine if the organization wants to deploy a number of large subnets, but what about the occasional small subnet containing only 20 or 30 hosts? Since a subnetted network could have only a single mask, the network administrator was still required to assign the 20 or 30 hosts to a subnet with a 22-bit prefix. This assignment would waste approximately 1,000 IP host addresses for each small subnet deployed! Limiting the association of a network number with a single mask did not encourage the flexible and efficient use of an organization's address space.

One solution to this problem was to allow a subnetted network to be assigned more than one subnet mask. Assume that in the previous example, the network administrator is also allowed to configure the 130.5.0.0/16 network with a /26 extended-network-prefix. Please refer to Figure 16. A /16 network address with a /26 extended-network prefix permits 1024 subnets (2^{10}), each of which supports a maximum of 62 hosts (2^6-2). The /26 prefix would be ideal for small subnets with less than 60 hosts, while the /22 prefix is well suited for larger subnets containing up to 1000 hosts.
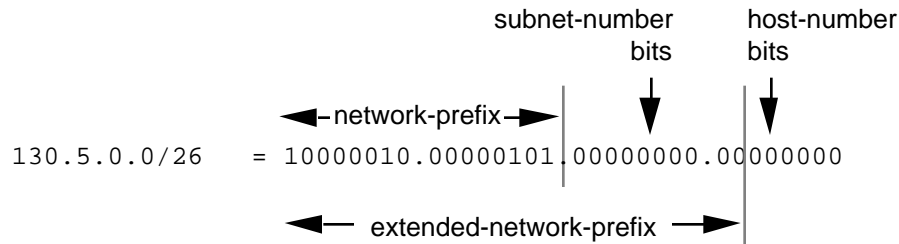
                                    subnet-number   host-number
                                    bits   bits

                    ◄─network-prefix─►   ▼   ▼
130.5.0.0/26   = 10000010.00000101.00000000.00000000

                    ◄─── extended-network-prefix ───►

Figure 16: 130.5.0.0/16 with a /26 Extended-Network Prefix

Route Aggregation

VLSM also allows the recursive division of an organization's address space so that it can
be reassembled and aggregated to reduce the amount of routing information at the top
level. Conceptually, a network is first divided into subnets, some of the subnets are
further divided into sub-subnets, and some of the sub-subnets are divided into sub[2]-
subnets. This allows the detailed structure of routing information for one subnet group
to be hidden from routers in another subnet group.
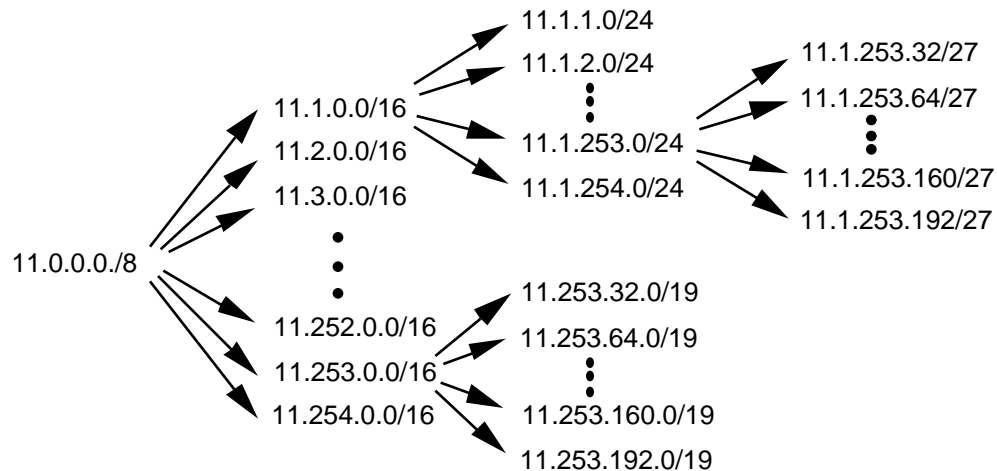
Figure 17: VLSM Permits the Recursive Division of a Network Prefix

In Figure 17, the 11.0.0.0/8 network is first configured with a /16 extended-network-
prefix. The 11.1.0.0/16 subnet is then configured with a /24 extended-network-prefix
and the 11.253.0.0/16 subnet is configured with a /19 extended-network-prefix. Note
that the recursive process does not require that the same extended-network-prefix be
assigned at each level of the recursion. Also, the recursive sub-division of the
organization's address space can be carried out as far as the network administrator needs
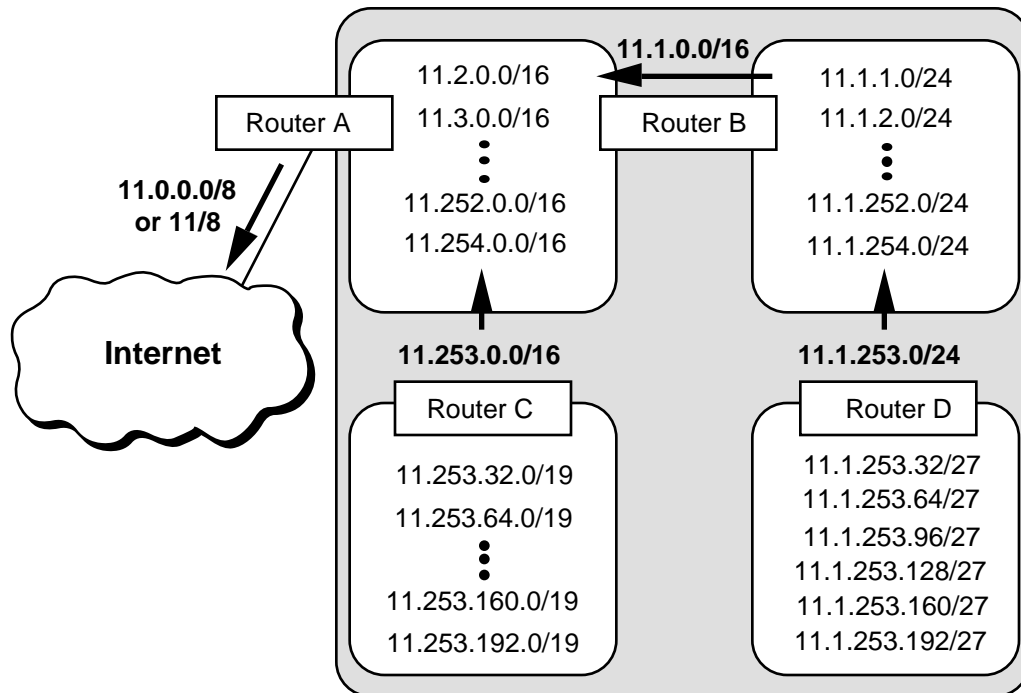to take it.

Figure 18: VLSM Permits Route Aggregation - Reducing Routing Table Size

Figure 18 illustrates how a planned and thoughtful allocation of VLSM can reduce the size of an organization's routing tables. Notice how Router D is able to summarize the six subnets behind it into a single advertisement (11.1.253.0/24) and how Router B is able to aggregate all of subnets behind it into a single advertisement. Likewise, Router C is able to summarize the six subnets behind it into a single advertisement (11.253.0.0/16). Finally, since the subnet structure is not visible outside of the organization, Router A injects a single route into the global Internet's routing table - 11.0.0.0/8 (or 11/8).

VLSM Design Considerations

When developing a VLSM design, the network designer must recursively ask the same set of questions as for a traditional subnet design. The same set of design decisions must be made at each level of the hierarchy:

1) How many total subnets does this level need today?

2) How many total subnets will this level need in the future?

3) How many hosts are there on this level's largest subnet today?

4) How many hosts will there be on this level's largest subnet be in the future?

At each level, the design team must make sure that they have enough extra bits to support the required number of sub-entities in the next and further levels of recursion.

Assume that a network is spread out over a number of sites. For example, if an organization has three campuses today it probably needs 3-bits of subnetting ($2^3 = 8$) to allow the addition of more campuses in the future. Now, within each campus, there is likely to be a secondary level of subnetting to identify each building. Finally, within each building, a third level of subnetting might identify each of the individual workgroups. Following this hierarchical model, the top level is determined by the number of campuses, the mid-level is based on the number of buildings at each site, and the lowest level is determined by the "maximum number of subnets/maximum number of users per subnet" in each building.

The deployment of a hierarchical subnetting scheme requires careful planning. It is essential that the network designers recursively work their way down through their addressing plan until they get to the bottom level. At the bottom level, they must make sure that the leaf subnets are large enough to support the required number of hosts. When the addressing plan is deployed, the addresses from each site will be aggregable into a single address block that keeps the backbone routing tables from becoming too large.

Requirements for the Deployment of VLSM

The successful deployment of VLSM has three prerequisites:

- The routing protocols must carry extended-network-prefix information with each route advertisement.

- All routers must implement a consistent forwarding algorithm based on the "longest match."

- For route aggregation to occur, addresses must be assigned so that they have topological significance.

Routing Protocols Must Carry Extended-Network-Prefix Lengths

Modern routing protocols, such as OSPF and I-IS-IS, enable the deployment of VLSM by providing the extended-network-prefix length or mask value along with each route advertisement. This permits each subnetwork to be advertised with its corresponding prefix length or mask. If the routing protocols did not carry prefix information, a router would have to either assume that the locally configured prefix length should be applied, or perform a look-up in a statically configured prefix table that contains all of the required masking information. The first alternative cannot guarantee that the correct prefix is applied, and static tables do not scale since they are difficult to maintain and subject to human error.

The bottom line is that if you want to deploy VLSM in a complex topology, you must select OSPF or I-IS-IS as the Interior Gateway Protocol (IGP) rather than RIP-1! It should be mentioned that RIP-2, defined in RFC 1388, improves the RIP protocol by allowing it to carry extended-network-prefix information. Therefore, RIP-2 supports the deployment of VLSM.

Forwarding Algorithm is Based on the "Longest Match"

All routers must implement a consistent forwarding algorithm based on the "longest match" algorithm. The deployment of VLSM means that the set of networks associated with extended-network-prefixes may manifest a subset relationship. A route with a longer extended-network-prefix describes a smaller set of destinations than the same route with a shorter extended-network-prefix. As a result, a route with a longer extended-network-prefix is said to be "more specific" while a route with a shorter extended-network-prefix is said to be "less specific." Routers must use the route with the longest matching extended-network-prefix (most specific matching route) when forwarding traffic.

For example, if a packet's destination IP address is 11.1.2.5 and there are three network prefixes in the routing table (11.1.2.0/24, 11.1.0.0/16, and 11.0.0.0/8), the router would select the route to 11.1.2.0/24. The 11.1.2.0/24 route is selected because its prefix has the greatest number of corresponding bits in the Destination IP address of the packet. This is illustrated in Figure 19.

```
   Destination     11.1.2.5    = 00001011.00000001.00000010.00000101

*  Route #1        11.1.2.0/24 = 00001011.00000001.00000010.00000000

   Route #2        11.1.0.0/16 = 00001011.00000001.00000000.00000000

   Route #3        11.0.0.0/8  = 00001011.00000000.00000000.00000000
```
Figure 19: Best Match is with the Route Having the Longest Prefix (Most Specific)

There is a very subtle but extremely important issue here. Since the destination address matches all three routes, it must be assigned to a host which is attached to the 11.1.2.0/24 subnet. If the 11.1.2.5 address is assigned to a host that is attached to the 11.1.0.0/16 or 11.0.0.0/8 subnet, the routing system will *never* route traffic to the host since the "longest match algorithm" assumes that the host is part of the 11.1.2.0/24 subnet. This means that great care must be taken when assigning host addresses to make sure that every host is reachable!

Topologically Significant Address Assignment

Since OSPF and I-IS-IS convey the extended-network-prefix information with each route, the VLSM subnets can be scattered throughout an organization's topology. However, to support hierarchical routing and reduce the size of an organization's routing tables, addresses should be assigned so that they are topologically significant.

Hierarchical routing requires that addresses be assigned to reflect the actual network topology. This reduces the amount of routing information by taking the set of addresses assigned to a particular region of the topology, and aggregating them into a single routing advertisement for the entire set. Hierarchical routing allows this to be done recursively at various points within the hierarchy of the routing topology. If addresses do not have a topological significance, aggregation cannot be performed and the size of the routing tables cannot be reduced. Remember this point when we discuss CIDR aggregation later in this paper.

VLSM Example

Given

An organization has been assigned the network number 140.25.0.0/16 and it plans to deploy VLSM. Figure 20 provides a graphic display of the VLSM design for the organization.
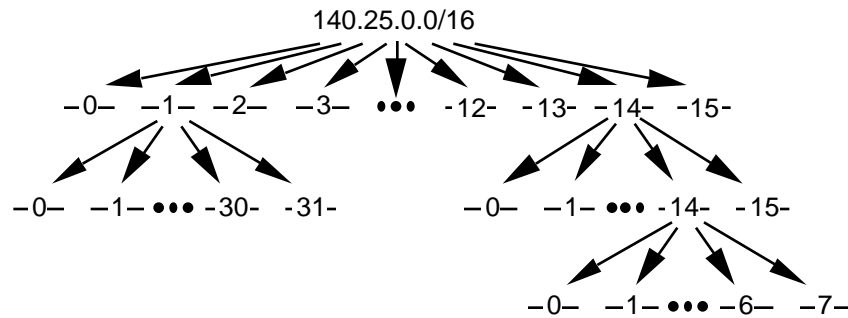

Figure 20: Address Strategy for VLSM Example

The first step of the subnetting process divides the base network address into 16 equal-sized address blocks. Then Subnet #1 is divided it into 32 equal-sized address blocks and Subnet #14 is divided into 16 equal-sized address blocks. Finally, Subnet #14-14 is divided into 8 equal-sized address blocks.

Define the 16 Subnets of 140.25.0.0/16

The first step in the subnetting process divides the base network address into 16 equal-size address blocks. This is illustrated in Figure 21.
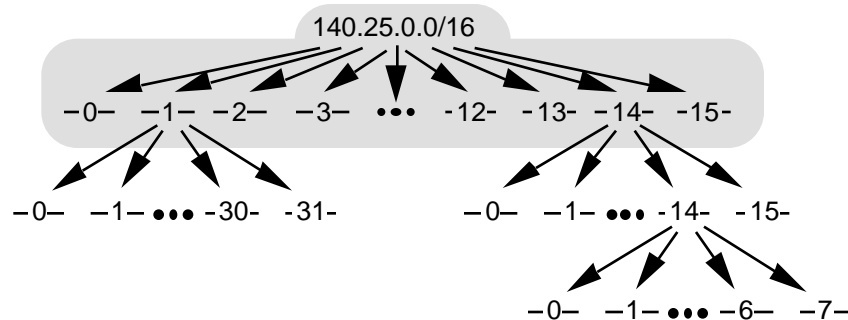

Figure 21: Define the 16 Subnets for 140.25.0.0/16

Since $16 = 2^4$, four bits are required to uniquely identify each of the 16 subnets. This means that the organization needs four more bits, or a /20, in the extended-network-prefix to define the 16 subnets of 140.25.0.0/16. Each of these subnets represents a contiguous block of 2^{12} (or 4,096) network addresses.

The 16 subnets of the 140.25.0.0/16 address block are given below. The subnets are numbered 0 through 15. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 4-bits representing the subnet-number field:

```
Base Network:   10001100.00011001.00000000.00000000 = 140.25.0.0/16

Subnet #0:      10001100.00011001.00000000.00000000 = 140.25.0.0/20
Subnet #1:      10001100.00011001.00010000.00000000 = 140.25.16.0/20
Subnet #2:      10001100.00011001.00100000.00000000 = 140.25.32.0/20
Subnet #3:      10001100.00011001.00110000.00000000 = 140.25.48.0/20
Subnet #4:      10001100.00011001.01000000.00000000 = 140.25.64.0/20
       :
       :
Subnet #13:     10001100.00011001.11010000.00000000 = 140.25.208.0/20
Subnet #14:     10001100.00011001.11100000.00000000 = 140.25.224.0/20
Subnet #15:     10001100.00011001.11110000.00000000 = 140.25.240.0/20
```

Define the Host Addresses for Subnet #3 (140.25.48.0/20)

Let's examine the host addresses that can be assigned to Subnet #3 (140.25.48.0/20). This is illustrated in Figure 22.
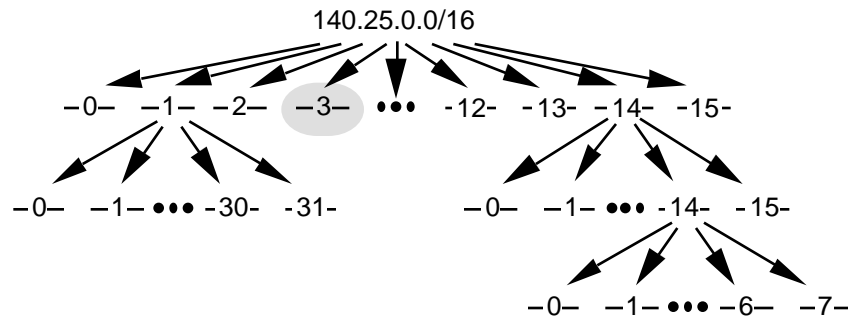

Figure 22: Define the Host Addresses for Subnet #3 (140.25.48.0/20)

Since the host-number field of Subnet #3 contains 12 bits, there are 4,094 valid host addresses (2^{12}-2) in the address block. The hosts are numbered 1 through 4,094.

The valid host addresses for Subnet #3 are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 12-bit host-number field:

```
Subnet #3:      10001100.00011001.00110000.00000000 = 140.25.48.0/20

Host #1:    10001100.00011001.00110000.00000001 = 140.25.48.1/20
Host #2:    10001100.00011001.00110000.00000010 = 140.25.48.2/20
Host #3:    10001100.00011001.00110000.00000011 = 140.25.48.3/20
   :
   :
Host #4093: 10001100.00011001.00111111.11111101 = 140.25.63.253/20
Host #4094: 10001100.00011001.00111111.11111110 = 140.25.63.254/20
```

The broadcast address for Subnet #3 is the all 1's host address or:

```
10001100.00011001.0011`1111.11111111` = 140.25.63.255
```

The broadcast address for Subnet #3 is exactly one less than the base address for Subnet #4 (140.25.64.0).

Define the Sub-Subnets for Subnet #14 (140.25.224.0/20)

After the base network address is divided into sixteen subnets, Subnet #14 is further subdivided into 16 equal-size address blocks. This is illustrated in Figure 23.
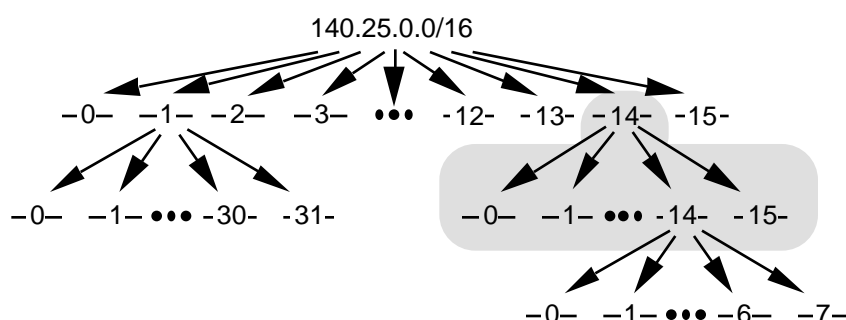

Figure 23: Define the Sub-Subnets for Subnet #14 (140.25.224.0/20)

Since $16 = 2^4$, four more bits are required to identify each of the 16 subnets. This means that the organization will need to use a /24 as the extended-network-prefix length.

The 16 subnets of the 140.25.224.0/20 address block are given below. The subnets are numbered 0 through 15. The underlined portion of each sub-subnet address identifies the extended-network-prefix, while the **bold** digits identify the 4-bits representing the sub-subnet-number field:

```
Subnet #14:    10001100.00011001.11100000.00000000 = 140.25.224.0/20

Subnet #14-0:  10001100.00011001.11100000.00000000 = 140.25.224.0/24
Subnet #14-1:  10001100.00011001.11100001.00000000 = 140.25.225.0/24
Subnet #14-2:  10001100.00011001.11100010.00000000 = 140.25.226.0/24
Subnet #14-3:  10001100.00011001.11100011.00000000 = 140.25.227.0/24
Subnet #14-4:  10001100.00011001.11100100.00000000 = 140.25.228.0/24
    :
    :
Subnet #14-14: 10001100.00011001.11101110.00000000 = 140.25.238.0/24
Subnet #14-15: 10001100.00011001.11101111.00000000 = 140.25.239.0/24
```

Define Host Addresses for Subnet #14-3 (140.25.227.0/24)

Let's examine the host addresses that can be assigned to Subnet #14-3 (140.25.227.0/24). This is illustrated in Figure 24.
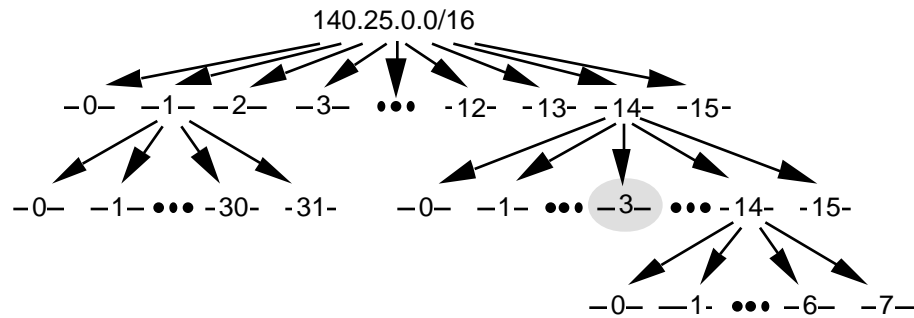

Figure 24: Define the Host Addresses for Subnet #14-3 (140.25.227.0/24)

Each of the subnets of Subnet #14-3 has 8 bits in the host-number field. This means that each subnet represents a block of 254 valid host addresses (2^8-2). The hosts are numbered 1 through 254.

The valid host addresses for Subnet #14-3 are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 8-bit host-number field:

```
Subnet #14-3:   10001100.00011001.11100011.00000000 = 140.25.227.0/24

Host #1        10001100.00011001.11100011.00000001 = 140.25.227.1/24
Host #2        10001100.00011001.11100011.00000010 = 140.25.227.2/24
Host #3        10001100.00011001.11100011.00000011 = 140.25.227.3/24
Host #4        10001100.00011001.11100011.00000100 = 140.25.227.4/24
Host #5        10001100.00011001.11100011.00000101 = 140.25.227.5/24
      .
      .
Host #253      10001100.00011001.11100011.11111101 = 140.25.227.253/24
Host #254      10001100.00011001.11100011.11111110 = 140.25.227.254/24
```

The broadcast address for Subnet #14-3 is the all 1's host address or:

```
      10001100.00011001.11100011.11111111 = 140.25.227.255
```

The broadcast address for Subnet #14-3 is exactly one less than the base address for Subnet #14-4 (140.25.228.0).

Define the Sub^2-Subnets for Subnet #14-14 (140.25.238.0/24)

After Subnet #14 was divided into sixteen subnets, Subnet #14-14 is further subdivided into 8 equal-size address blocks. This is illustrated in Figure 25.
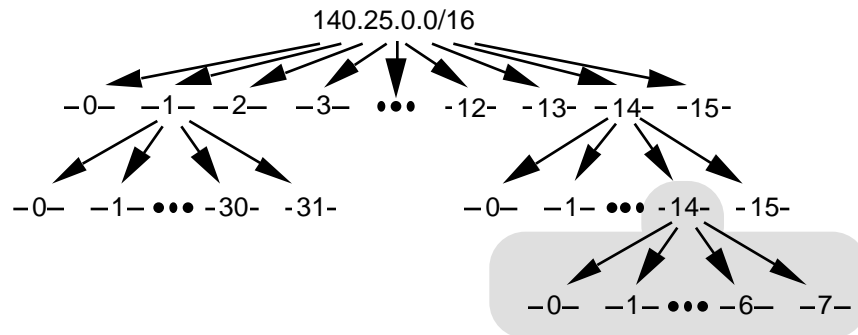

Figure 25: Define the Sub^2-Subnets for Subnet #14-14 (140.25.238.0/24)

Since $8 = 2^3$, three more bits are required to identify each of the 8 subnets. This means that the organization will need to use a /27 as the extended-network-prefix length.

The 8 subnets of the 140.25.238.0/24 address block are given below. The subnets are numbered 0 through 7. The underlined portion of each sub-subnet address identifies the extended-network-prefix, while the **bold** digits identify the 3-bits representing the subnet^2-number field:

```
Subnet #14-14: 10001100.00011001.11101110.00000000 = 140.25.238.0/24


Subnet#14-14-0:10001100.00011001.11101110.00000000 = 140.25.238.0/27
Subnet#14-14-1:10001100.00011001.11101110.00100000 = 140.25.238.32/27
Subnet#14-14-2:10001100.00011001.11101110.01000000 = 140.25.238.64/27
Subnet#14-14-3:10001100.00011001.11101110.01100000 = 140.25.238.96/27
Subnet#14-14-4:10001100.00011001.11101110.10000000 = 140.25.238.128/27
Subnet#14-14-5:10001100.00011001.11101110.10100000 = 140.25.238.160/27
Subnet#14-14-6:10001100.00011001.11101110.11000000 = 140.25.238.192/27
Subnet#14-14-7:10001100.00011001.11101110.11100000 = 140.25.238.224/27
```

Define Host Addresses for Subnet #14-14-2 (140.25.238.64/27)

Let's examine the host addresses that can be assigned to Subnet #14-14-2 (140.25.238.64/27). This is illustrated in Figure 26.
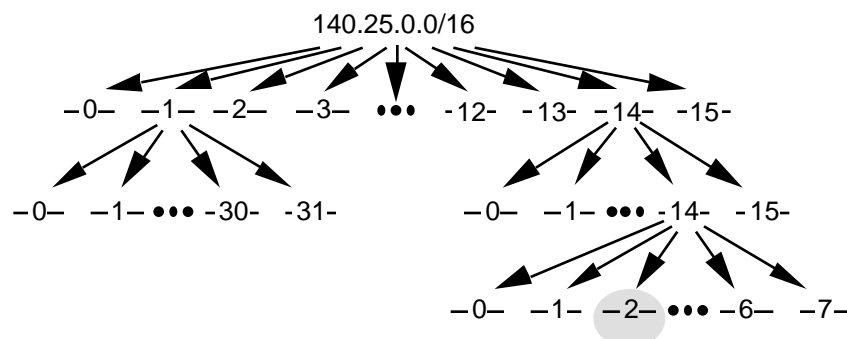
Figure 26: Define the Host Addresses for Subnet #14-14-2 (140.25.238.64/27)

Each of the subnets of Subnet #14-14 has 5 bits in the host-number field. This means that each subnet represents a block of 30 valid host addresses (2^5-2). The hosts will be numbered 1 through 30.

The valid host addresses for Subnet #14-14-2 are given below. The underlined portion of each address identifies the extended-network-prefix, while the **bold** digits identify the 5-bit host-number field:

```
Subnet#14-14-2:10001100.00011001.11101110.01000000 = 140.25.238.64/27

  Host #1      10001100.00011001.11101110.01000001 = 140.25.238.65/27
  Host #2      10001100.00011001.11101110.01000010 = 140.25.238.66/27
  Host #3      10001100.00011001.11101110.01000011 = 140.25.238.67/27
  Host #4      10001100.00011001.11101110.01000100 = 140.25.238.68/27
  Host #5      10001100.00011001.11101110.01000101 = 140.25.238.69/27

      .
      .
  Host #29     10001100.00011001.11101110.01011101 = 140.25.238.93/27
  Host #30     10001100.00011001.11101110.01011110 = 140.25.238.94/27
```

The broadcast address for Subnet #14-14-2 is the all 1's host address or:

```
      10001100.00011001.11011100.01011111 = 140.25.238.95
```

The broadcast address for Subnet #6-14-2 is exactly one less than the base address for Subnet #14-14-3 (140.25.238.96).

Additional Practice with VLSM

Please turn to Appendix D for practice exerciss to reinforce your understanding of VLSM.

Classless Inter-Domain Routing (CIDR)

By 1992, the exponential growth of the Internet was beginning to raise serious concerns among members of the IETF about the ability of the Internet's routing system to scale and support future growth. These problems were related to:

- The near-term exhaustion of the Class B network address space
- The rapid growth in the size of the global Internet's routing tables
- The eventual exhaustion of the 32-bit IPv4 address space

Projected Internet growth figures made it clear that the first two problems were likely to become critical by 1994 or 1995. The response to these immediate challenges was the development of the concept of Supernetting or Classless Inter-Domain Routing (CIDR). The third problem, which is of a more long-term nature, is currently being explored by the IP Next Generation (IPng or IPv6) working group of the IETF.

CIDR was officially documented in September 1993 in RFC 1517, 1518, 1519, and 1520. CIDR supports two important features that benefit the global Internet routing system:

- CIDR eliminates the traditional concept of Class A, Class B, and Class C network addresses. This enables the efficient allocation of the IPv4 address space which will allow the continued growth of the Internet until IPv6 is deployed.

- CIDR supports route aggregation where a single routing table entry can represent the address space of perhaps thousands of traditional classful routes. This allows a single routing table entry to specify how to route traffic to many individual network addresses. Route aggregation helps control the amount of routing information in the Internet's backbone routers, reduces route flapping (rapid changes in route availability), and eases the local administrative burden of updating external routing information.

Without the rapid deployment of CIDR in 1994 and 1995, the Internet routing tables would have in excess of 70,000 routes (instead of the current 30,000+) and the Internet would probably not be functioning today!

CIDR Promotes the Efficient Allocation of the IPv4 Address Space

CIDR eliminates the traditional concept of Class A, Class B, and Class C network addresses and replaces them with the generalized concept of a "network-prefix." Routers use the network-prefix, rather than the first 3 bits of the IP address, to determine the dividing point between the network number and the host number. As a result, CIDR supports the deployment of *arbitrarily sized* networks rather than the standard 8-bit, 16-bit, or 24-bit network numbers associated with classful addressing.

In the CIDR model, each piece of routing information is advertised with a bit mask (or prefix-length). The prefix-length is a way of specifying the number of leftmost contiguous bits in the network-portion of each routing table entry. For example, a

network with 20 bits of network-number and 12-bits of host-number would be advertised with a 20-bit prefix length (a /20). The clever thing is that the IP address advertised with the /20 prefix could be a former Class A, Class B, or Class C. Routers that support CIDR do *not* make assumptions based on the first 3-bits of the address, they rely on the prefix-length information provided with the route.

In a classless environment, prefixes are viewed as bitwise contiguous blocks of the IP address space. For example, all prefixes with a /20 prefix represent the same amount of address space (2^{12} or 4,096 host addresses). Furthermore, a /20 prefix can be assigned to a traditional Class A, Class B, or Class C network number. Figure 27 shows how each of the following /20 blocks represent 4,096 host addresses - 10.23.64.0/20, 130.5.0.0/20, and 200.7.128.0/20.

```
Traditional A    10.23.64.0/20     00001010.00010111.01000000.00000000

Traditional B    130.5.0.0/20      10000010.00000101.00000000.00000000

Traditional C    200.7.128.0/20    11001000.00000111.10000000.00000000
```
Figure 27: /20 Bitwise Contiguous Address Blocks

Table 3 provides information about the most commonly deployed CIDR address blocks. Referring to the Table, you can see that a /15 allocation can also be specified using the traditional dotted-decimal mask notation of 255.254.0.0. Also, a /15 allocation contains a bitwise contiguous block of 128K (131,072) IP addresses which can be classfully interpreted as 2 Class B networks or 512 Class C networks.

Table 3: CIDR Address Blocks

CIDR prefix-length	Dotted-Decimal	# Individual Addresses	# of Classful Networks
/13	255.248.0.0	512 K	8 Bs or 2048 Cs
/14	255.252.0.0	256 K	4 Bs or 1024 Cs
/15	255.254.0.0	128 K	2 Bs or 512 Cs
/16	255.255.0.0	64 K	1 B or 256 Cs
/17	255.255.128.0	32 K	128 Cs
/18	255.255.192.0	16 K	64 Cs
/19	255.255.224.0	8 K	32 Cs
/20	255.255.240.0	4 K	16 Cs
/21	255.255.248.0	2 K	8 Cs
/22	255.255.252.0	1 K	4 Cs
/23	255.255.254.0	512	2 Cs
/24	255.255.255.0	256	1 C
/25	255.255.255.128	128	1/2 C
/26	255.255.255.192	64	1/4 C
/27	255.255.255.224	32	1/8 C

Host Implications for CIDR Deployment

It is important to note that there may be severe host implications when you deploy CIDR based networks. Since many hosts are classful, their user interface will not permit them to be configured with a mask that is shorter than the "natural" mask for a traditional classful address. For example, potential problems could exist if you wanted to deploy 200.25.16.0 as a /20 to define a network capable of supporting 4,094 (2^{12}-2) hosts. The software executing on each end station might not allow a traditional Class C (200.25.16.0) to be configured with a 20-bit mask since the natural mask for a Class C network is a 24-bit mask. If the host software supports CIDR, it will permit shorter masks to be configured.

However, there will be no host problems if you were to deploy the 200.25.16.0/20 (a traditional Class C) allocation as a block of 16 /24s since non-CIDR hosts will interpret their local /24 as a Class C. Likewise, 130.14.0.0/16 (a traditional Class B) could be deployed as a block of 255 /24s since the hosts will interpret the /24s as subnets of a /16. If host software supports the configuration of shorter than expected masks, the network manager has tremendous flexibility in network design and address allocation.

Efficient Address Allocation

How does all of this lead to the efficient allocation of the IPv4 address space? In a classful environment, an Internet Service Provider (ISP) can only allocate /8, /16, or /24 addresses. In a CIDR environment, the ISP can carve out a block of its registered address space that specifically meets the needs of each client, provides additional room for growth, and does not waste a scarce resource.

Assume that an ISP has been assigned the address block 206.0.64.0/18. This block represents 16,384 (2^{14}) IP addresses which can be interpreted as 64 /24s. If a client requires 800 host addresses, rather than assigning a Class B (and wasting ~64,700 addresses) or four individual Class Cs (and introducing 4 new routes into the global Internet routing tables), the ISP could assign the client the address block 206.0.68.0/22, a block of 1,024 (2^{10}) IP addresses (4 contiguous /24s). The efficiency of this allocation is illustrated in Figure 28.

```
ISP's Block:   11001110.00000000.01000000.00000000    206.0.64.0/18


Client Block:  11001110.00000000.01000100.00000000    206.0.68.0/22

Class C #0:    11001110.00000000.01000100.00000000    206.0.68.0/24

Class C #1:    11001110.00000000.01000101.00000000    206.0.69.0/24

Class C #2:    11001110.00000000.01000110.00000000    206.0.70.0/24

Class C #3:    11001110.00000000.01000111.00000000    206.0.71.0/24
```

Figure 28: CIDR Supports Efficient Address Allocation

CIDR Address Allocation Example

For this example, assume that an ISP owns the address block 200.25.0.0/16. This block represents 65, 536 (2^{16}) IP addresses (or 256 /24s).

From the 200.25.0.0/16 block it wants to allocate the 200.25.16.0/20 address block . This smaller block represents 4,096 (2^{12}) IP addresses (or 16 /24s).

```
Address Block   11001000.00011001.00010000.00000000   200.25.16.0/20
```

In a classful environment, the ISP is forced to use the /20 as 16 individual /24s.

```
Network #0      11001000.00011001.00010000.00000000   200.25.16.0/24
Network #1      11001000.00011001.00010001.00000000   200.25.17.0/24
Network #2      11001000.00011001.00010010.00000000   200.25.18.0/24
Network #3      11001000.00011001.00010011.00000000   200.25.19.0/24
Network #4      11001000.00011001.00010100.00000000   200.25.20.0/24
                    :
                    :
Network #13     11001000.00011001.00011101.00000000   200.25.29.0/24
Network #14     11001000.00011001.00011110.00000000   200.25.30.0/24
Network #15     11001000.00011001.00011111.00000000   200.25.31.0/24
```

If you look at the ISP's /20 address block as a pie, in a classful environment it can only be cut into 16 equal-size pieces. This is illustrated in Figure 29.
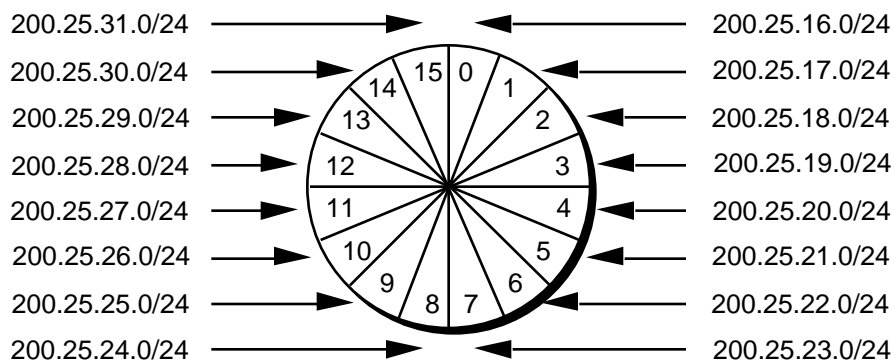

Figure 29: Slicing the Pie - Classful Environment

However, in a classless environment, the ISP is free to cut up the pie any way it wants. It could slice up the original pie into 2 pieces (each 1/2 of the address space) and assign one portion to Organization A, then cut the other half into 2 pieces (each 1/4 of the address space) and assign one piece to Organization B, and finally slice the remaining fourth into 2 pieces (each 1/8 of the address space) and assign it to Organization C and Organization D. Each of the individual organizations is free to allocate the address space within its "Intranetwork" as it sees fit. This is illustrated in Figure 30.
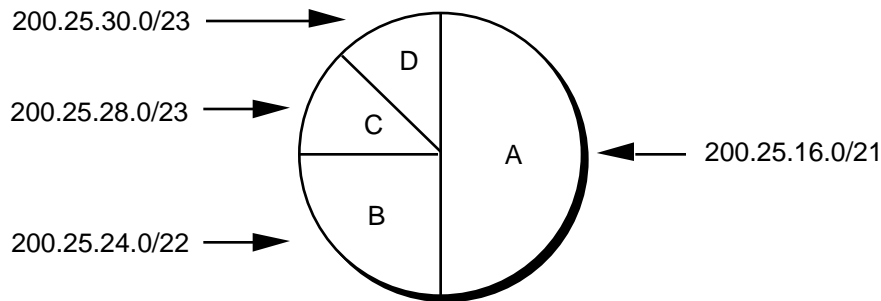
200.25.30.0/23

200.25.28.0/23

200.25.24.0/22

200.25.16.0/21

Figure 30: Slicing the Pie - Classless Environment

Step #1: Divide the address block 200.25.16.0/20 into two equal size slices. Each block represents one-half of the address space or 2,048 (2^{11}) IP addresses.

```
ISP's Block   11001000.00011001.0001 0000.00000000   200.25.16.0/20

Org A:        11001000.00011001.0001 0000.00000000   200.25.16.0/21
Reserved:     11001000.00011001.0001 1000.00000000   200.25.24.0/21
```

Step #2: Divide the reserved block (200.25.24.0/21) into two equal size slices. Each block represents one-fourth of the address space or 1,024 (2^{10}) IP addresses.

```
Reserved      11001000.00011001.00011 000.00000000   200.25.24.0/21

Org B:        11001000.00011001.00011 000.00000000   200.25.24.0/22
Reserved      11001000.00011001.00011 100.00000000   200.25.28.0/22
```

Step #3: Divide the reserved address block (200.25.28.0/22) into two equal size blocks. Each block represents one-eight of the address space or 512 (2^9) IP addresses.

```
Reserved      11001000.00011001.000111 00.00000000   200.25.28.0/22

Org C:        11001000.00011001.000111 00.00000000   200.25.28.0/23
Org D:        11001000.00011001.000111 10.00000000   200.25.30.0/23
```

CIDR is Similar to VLSM

If CIDR appears to have the familiar look and feel of VLSM, you're correct! CIDR and VLSM are essentially the same thing since they both allow a portion of the IP address space to be recursively divided into subsequently smaller pieces. The difference is that with VLSM, the recursion is performed on the address space previously assigned to an organization and is invisible to the global Internet. CIDR, on the other hand, permits the recursive allocation of an address block by an Internet Registry to a high-level ISP, to a mid-level ISP, to a low-level ISP, and finally to a private organization's network.

Just like VLSM, the successful deployment of CIDR has three prerequisites:

- The routing protocols must carry network-prefix information with each route advertisement.

- All routers must implement a consistent forwarding algorithm based on the "longest match."

- For route aggregation to occur, addresses must be assigned so that they are topologically significant.

Controlling the Growth of Internet's Routing Tables

Another important benefit of CIDR is that it plays an important role in controlling the growth of the Internet's routing tables. The reduction of routing information requires that the Internet be divided into addressing domains. Within a domain, detailed information is available about all of the networks that reside in the domain. Outside of an addressing domain, only the common network prefix is advertised. This allows a single routing table entry to specify a route to many individual network addresses.
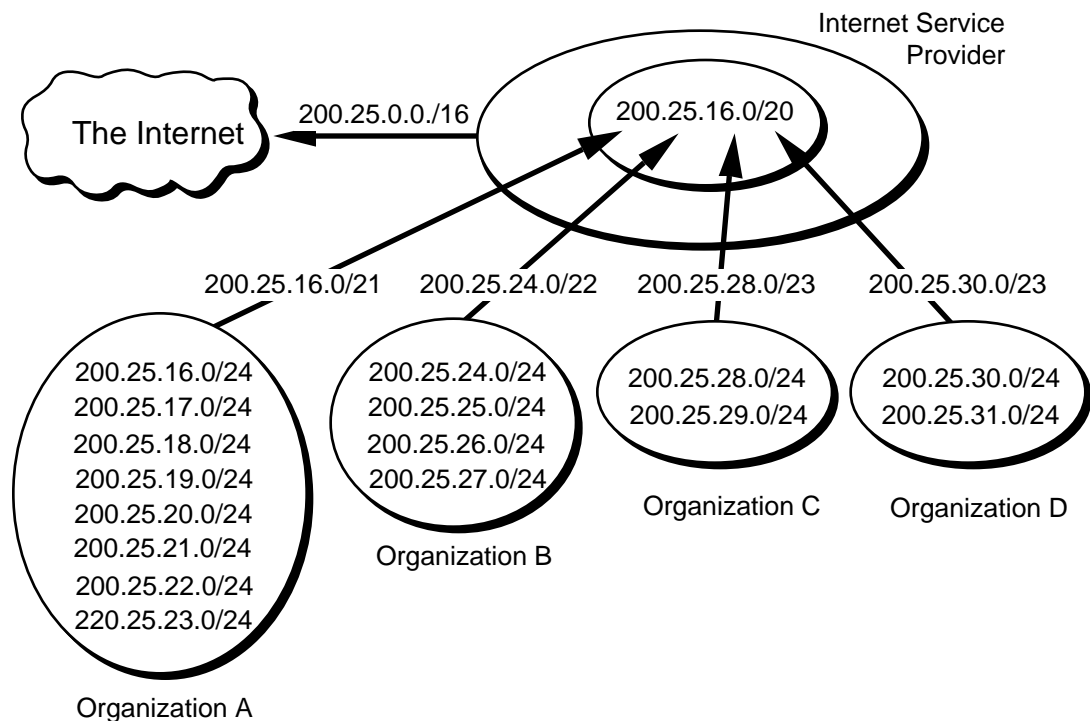

Figure 31: CIDR Reduces the Size of Internet Routing Tables

Figure 31 illustrates how the allocation described in previous CIDR example helps reduce the size of the Internet routing tables. Assume that a portion of the ISPs address block (200.25.16.0/20) has been allocated as described in the previous example. Organization A aggregates 8 /24s into a single advertisement (200.25.16.0/21), Organization B aggregates 4 /24s into a single advertisement (200.25.24.0/22), Organization C aggregates 2 /24s into a single advertisement (200.25.28.0/23), and

Organization D aggregates 2 /24s into a single advertisement (200.25.30.0/23). Finally, the ISP is able to inject the 256 /24s in its allocation into the Internet with a single advertisement - 200.25.0.0/16!

It should be mentioned that route aggregation via BGP-4 is not automatic. The network engineers must configure each router to perform the required aggregation. The successful deployment of CIDR will allow the number of individual networks on the Internet to expand, while minimizing the number of routes in the Internet routing tables.

Routing in a Classless Environment

Figure 32 illustrates the routing advertisements for Organization A discussed in the previous CIDR Example.
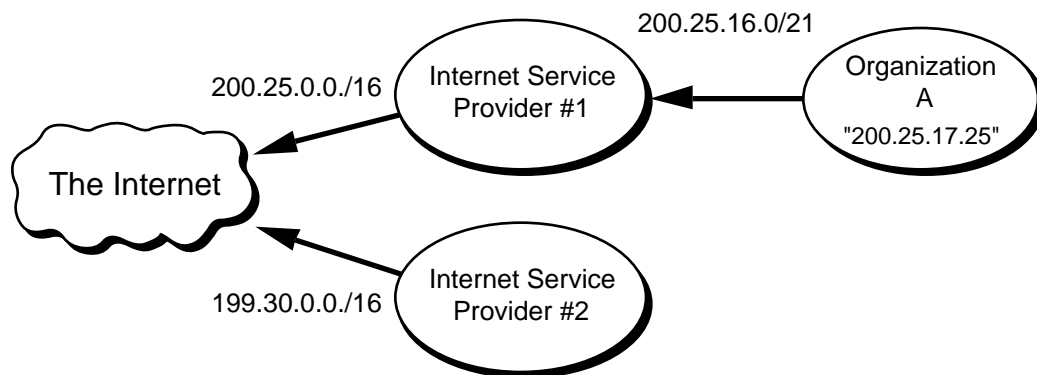

Figure 32: Routing Advertisements for Organization A

Since all of Organization A's routes are part of ISP #1's address block, the routes to Organization A are implicitly aggregated via ISP #1's aggregated announcement to the Internet. In other words, the eight networks assigned to Organization A are hidden behind a single routing advertisement. Using the longest match forwarding algorithm, Internet routers will route traffic to host 200.25.17.25 to ISP #1, which will in turn route the traffic to Organization A.

Now, for whatever reasons, assume that Organization A decides to change its network provider to ISP #2. This is illustrated in Figure 33.
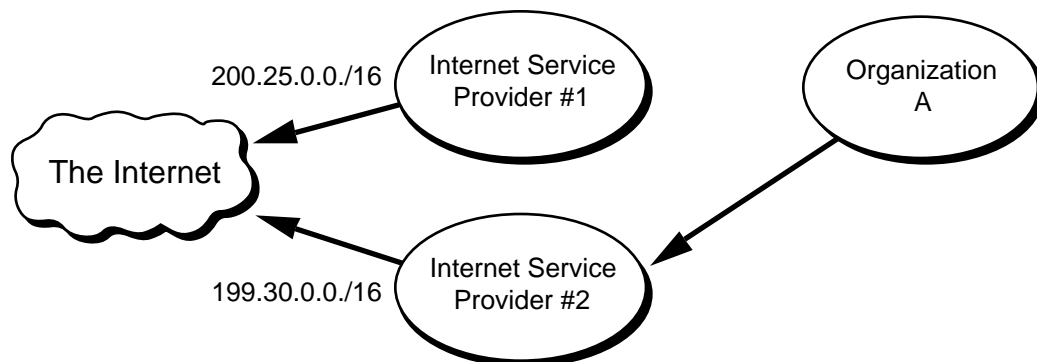

Figure 33: Organization A Changes Network Providers to ISP #2

The "best" thing for the size of the Internet's routing tables would be to have Organization A obtain a block of ISP #2's address space and renumber. This would allow the eight networks assigned to Organization A to be hidden behind the aggregate routing advertisement of ISP #2. Unfortunately, renumbering is a labor-intensive task which could be very difficult, if not impossible, for Organization A.
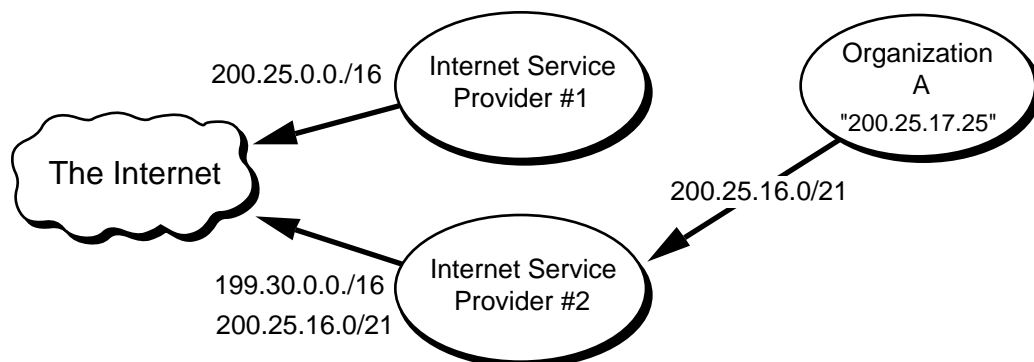

Figure 34: ISP #2 Injects a More-Specific Route into the Internet

The "best" thing for Organization A is to retain ownership of its address space and have ISP #2 advertise an "exception" (more specific) route into the Internet. The exception route allows all traffic for 200.25.0.0/16 to be sent to ISP #1, with the exception of the traffic to 200.25.16.0/21. This is accomplished by having ISP #2 advertise, in addition to its own 199.30.0.0/16 block, a route for 200.25.16.0/21. Please refer to Figure 34. Using the "longest match" forwarding algorithm, Internet routers will route traffic addressed to host 200.25.17.25 to ISP #2 which will in turn route the traffic to Organization A. Clearly, the introduction of a large number of exception routes can reduce the effectiveness of the CIDR deployment and eventually cause Internet routing tables to begin exploding again!

NETBuilder Support for CIDR

Support for CIDR has been implemented on the NETBuilder:

- NETBuilder software implements BGP-4. Support for CIDR is a significant part of the improvements made to BGP-4.

- NETBuilder software uses a routing table structure that understands a network number advertised with a prefix that is shorter than the natural mask. The NETBuilder's routing table and forwarding process ignore the traditional IP address Class and are capable of accepting any network/mask combination that it receives.

- NETBuilder software is capable of performing aggregation by way of BGP-4 configuration parameters. Also, the OSPF AreaRange parameter allows VLSM-based aggregation to be performed within an autonomous system. The network administrator may specify exactly what network numbers and masks are advertised outside of each area or domain.

Additional Practice with CIDR

Please turn to Appendix E for several practice exercises to reinforce your understanding of CIDR.

New Solutions for Scaling the Internet Address Space

As we approach the turn of the century, the problems of IPv4 address shortages and expanding Internet routing tables are still with us. The good news is that CIDR is working. The bad news is that recent growth trends indicate that the number of Internet routes is beginning to, once again, increase at an exponential rate. The Internet must find a way to keep the routing table growth linear. The IETF is continuing its efforts to develop solutions that will overcome these problems, enabling the continued growth and scalability of the Internet.

Appeal to Return Unused IP Network Prefixes

RFC 1917 requests that the Internet community return unused address blocks to the Internet Assigned Numbers Authority (IANA) for redistribution. This includes unused network numbers, addresses for networks that will never be connected to the global Internet for security reasons, and sites that are using a small percentage of their address space. RFC 1917 also petitions ISPs to return unused network-prefixes that are outside of their assigned address blocks. It will be interesting to see how the Internet community responds since many organizations with unused addresses don't want to return them because they are viewed as an asset.

Address Allocation for Private Internets

RFC 1918 requests that organizations make use of the private Internet address space for hosts that require IP connectivity within their enterprise network, but do not require external connections to the global Internet. For this purpose, the IANA has reserved the following three address blocks for private internets:

```
10.0.0.0    - 10.255.255.255      (10/8 prefix)
172.16.0.0  - 172.31.255.255      (172.16/12 prefix)
192.168.0.0 - 192.168.255.255     (192.168/16 prefix)
```

Any organization that elects to use addresses from these reserved blocks can do so without contacting the IANA or an Internet registry. Since these addresses are never injected into the global Internet routing system, the address space can simultaneously be used by many different organizations.

The disadvantage to this addressing scheme is that it requires an organization to use a Network Address Translator (NAT) for global Internet access. However, the use of the private address space and a NAT make it much easier for clients to change their ISP without the need to renumber or "punch holes" in a previously aggregated advertisement. The benefits of this addressing scheme to the Internet is that it reduces the demand for IP addresses so large organizations may require only a small block of the globally unique IPv4 address space.

Address Allocation from the Reserved Class A Address Space

An Internet draft, "Observations on the use of Components of the Class A Address Space within the Internet" <draft-ietf-cidrd-classa-01.txt>, explores the allocation of the upper-half of the currently reserved Class A address space through delegated registries. As the demand for IP addresses continues to grow, it appears that it may be necessary to eventually allocate the 64.0.0.0/2 address space. Note that the 64.0.0.0/2 address block is huge and represents 25% of the IPv4 unicast address space.

Implications of Address Allocation Policies

An Internet draft , "Implications of Various Address Allocation Policies for Internet Routing" <draft-ietf-cidrd-addr-ownership-07.txt>, discusses the fundamental issues that must be considered as the Internet develops a new unicast address allocation and management policies. The draft compares the benefits and limitations of an "address ownership" policy with an "address lending" policy.

"Address ownership" means that when an address block is assigned to an organization, it remains allocated to that organization for as long as the organization wants to keep it. This means that the address block is "portable" and that the organization would be able to use it to gain access to the Internet no matter where the organization connects to the Internet. On the other hand, "address lending" means that an organization obtains its address block on a "loan" basis. If the loan ends, the organization can no longer use the borrowed address block, must obtain new addresses, and renumber before using them.

As we have seen, hierarchical routing requires that addresses reflect the network topology in order to permit route aggregation. The draft argues that there are two fundamental problems that break the hierarchical addressing and routing model supported by CIDR:

- The continued existence of pre-CIDR routes that cannot be aggregated.

- Organizations that switch ISPs and continue to use addresses from their previous ISP's address block. The new ISP cannot aggregate the old address block as part of its aggregation, so it must inject an exception route into the Internet. If the number of exception routes continues to increases, they will erode the benefits of CIDR and prevent the scalability of the Internet's routing system.

The draft concludes with the recommendation that large providers, which can express their destinations with a single prefix, be assigned address blocks following the "address ownership" model. However, all allocations from these providers to a downstream clients should follow the "address lending" model. This means that if an organization changes its provider, the loan is canceled and the client will be required to renumber.

This draft has generated a tremendous amount of discussion within the Internet community about the concept of address ownership and what it means in the context of global routing. The authors present a strong argument that the Internet has to make a choice between either address ownership for all or a routable Internet - it can't have

both! Smaller organizations that want to own their addresses have concerns about the difficulty of renumbering and their lack of self-determination if their provider or their provider's upstream provider changes its provider. Finally, ISPs have concerns because the term "large provider" has not been defined. At this time, the discussion continues since any criteria recommended by the IETF is bound to be perceived as unfair by some!

Procedures for Internet/Enterprise Renumbering (PIER)

In the face of the "address ownership" vs. "address lending" debate, it is clear that renumbering may become a critical issue in the late 1990s. Procedures for Internet/Enterprise Renumbering (PIER) is a working group of the IETF charged with the task of developing a renumbering strategy.

RFC 1916 is a request by PIER for the Internet community to provide assistance in the development of a series of documents describing how an organization might proceed to renumber its network. The ultimate goal of these documents is to provide education and practical experience to the Internet community.

Market-Based Allocation of IP Address Blocks

An Internet draft ,"Suggestions for Market-Based Allocation of IP Address Blocks" <draft-ietf-cidrd-blocks-00.txt>, is a proposal to make IPv4 address assignments transferable and condones the exchange of money as part of the transfer procedure. It suggests that the Internet community embrace the profit motive as an incentive to motivate organizations to act in ways that will improve resource use. This proposal goes hand-in-hand with another proposal to introduce financial incentives for route aggregation (i.e., have ISPs levy a charge for each route advertised). The idea is to move the decisions regarding scarce resources from a political atmosphere to a financial environment which is better suited to deal with scarcity.

Keeping Current on Internet Addressing Issues

General Internet Information

Internet Monthly Reports discuss the accomplishments, milestones, and problems discovered on the Internet. They are available from: <http://info.internet.isi.edu/1/in-notes/imr>

Minutes of the most recent IETF Proceedings are available from: <http://www.ietf.cnri.reston.va.us/proceedings/directory.html>

Information about the size and content of the Internet routing table is available on the Merit Web pages: <http://www.ra.net/~ra/statistics/routes.html>

CIDR Deployment (CIDRD)

For general information about the CIDRD working group of the IETF and its charter: <http://www.ietf.cnri.reston.va.us/html.charters/cidrd-charter.html>

To subscribe to the CIDRD mailing list: <cidrd-request@iepg.org>

The CIDRD mail list archive: <ftp://aarnet.edu.au/pub/mailing-lists/cidrd>

Internet Drafts published by the CIDRD working group are available from: <http://www.ietf.cnri.reston.va.us/ids.by.wg/cidrd.html>

Procedures for Internet/Enterprise Renumbering (PIER)

General information about the PIER working group of the IETF and its charter is available from: <http://www.ietf.cnri.reston.va.us/html.charters/pier-charter.html>

To subscribe to the PIER mailing list: <pier-request@isi.edu>

The PIER mail list archive: < ftp://ftp.isi.edu/pier-archive>

Papers developed by PIER are available from: <http://www.isi.edu:80/div7/pier/>.

Dynamic Host Configuration (DHCP)

For information about the DHCP working group, current Internet-Drafts, and Requests for Comments: <http://www.ietf.cnri.reston.va.us/html.charters/dhc-charter.html>

To access the DHCP Home Page: <http://charlotte.acns.nwu.edu/internet/tech/dhcp/>

To subscribe to the DHCP mailing list: <host-conf-request@sol.eg.bucknell.edu>

The DHCP mail list archive: <ftp://ftp.bucknell.edu/pub/dhcp>

IPng (IPNGWG)

For information about the IPng working group, current Internet-Drafts, and Requests for Comments: <http://www.ietf.cnri.reston.va.us/html.charters/ipngwg-charter.html>

To access the IPng Home Page: <http://playground.sun.com/pub/ipng/html/ipng-main.html>

To subscribe to the IPng mailing list: < majordomo@sunroof.eng.sun.com>

The IPng mail list archive: <ftp://parcftp.xerox.com/pub/ipng>

Appendix A - References

Requests for Comments

Requests for Comments are available on the WWW from: <http://ds.internic.net/ds/dspg2intdoc.html>

950   J. Mogul, J. Postel, "Internet standard subnetting procedure", 08/01/1985. (Pages=18) (STD 5)

985   National Science Foundation, Network Technical Advisory Group, "Requirements for Internet gateways - draft", 05/01/1986. (Pages=23) (Obsoleted by RFC1009)

1009   R. Braden, J. Postel, "Requirements for Internet gateways", 06/01/1987. (Pages=55) (Obsoletes RFC985) (STD 4) (Obsoleted by RFC1716)

1245   J. Moy, "OSPF Protocol Analysis", 08/08/1991. (Pages=12)

1246   J. Moy, "Experience with the OSPF Protocol", 08/08/1991. (Pages=31)

1247   J. Moy, "OSPF Version 2", 08/08/1991. (Pages=189) (Format=.txt, .ps) (Obsoletes RFC1131) (Obsoleted by RFC1583)

1338   V. Fuller, T. Li, K. Varadhan, J. Yu, "Supernetting: an Address Assignment and Aggregation Strategy", 06/26/1992. (Pages=20) (Obsoleted by RFC1519)

1366   E. Gerich, "Guidelines for Management of IP Address Space", 10/22/1992. (Pages=8) (Obsoleted by RFC1466)

1466   E. Gerich, "Guidelines for Management of IP Address Space", 05/26/1993. (Pages=10) (Obsoletes RFC1366)

1517   R. Hinden, "Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR)", 09/24/1993. (Pages=4)

1518   Y. Rekhter, T. Li, "An Architecture for IP Address Allocation with CIDR", 09/24/1993. (Pages=27)

1519   V. Fuller, T. Li, J. Yu, K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", 09/24/1993. (Pages=24) (Obsoletes RFC1338)

1520   Y. Rekhter, C. Topolcic, "Exchanging Routing Information Across Provider Boundaries in the CIDR Environment", 09/24/1993. (Pages=9)

1583   J. Moy, "OSPF Version 2", 03/23/1994. (Pages=212) (Obsoletes RFC1247)

1716   P. Almquist, F. Kastenholz, "Towards Requirements for IP Routers", 11/04/1994. (Pages=186) (Obsoletes RFC1009) (Obsoleted by RFC1812)

1721   G. Malkin, "RIP Version 2 Protocol Analysis", 11/15/1994. (Pages=4) (Obsoletes RFC1387)

1722   G. Malkin, "RIP Version 2 Protocol Applicability Statement", 11/15/1994. (Pages=5)

1723   G. Malkin, "RIP Version 2 Carrying Additional Information", 11/15/1994. (Pages=9) (Updates RFC1058) (Obsoletes RFC1388)

1724   G. Malkin, F. Baker, "RIP Version 2 MIB Extension", 11/15/1994. (Pages=18) (Obsoletes RFC1389)

1812   F. Baker, "Requirements for IP Version 4 Routers", 06/22/1995. (Pages=175) (Obsoletes RFC1716)

1900   B. Carpenter, Y. Rekhter, "Renumbering Needs Work", 02/28/1996. (Pages=4)

1916   H. Berkowitz, P. Ferguson, W. Leland, P. Nesser, "Enterprise Renumbering: Experience and Information Solicitation", 02/28/1996. (Pages=8)

1917   P. Nesser, "An Appeal to the Internet Community to Return Unused IP Network (Prefixes) to the IANA", 02/29/1996. (Pages=10)

1918   Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot, E. Lear, , "Address Allocation for Private Internets", 02/29/1996. (Pages=9) (Obsoletes RFC1627)

Internet Drafts

Internet Drafts are available on the WWW from: <http://www.ietf.cnri.reston.va.us/1id-abstracts.html>

"Suggestions for Market-Based Allocation of IP Address Blocks", <draft-ietf-cidrd-blocks-00.txt>, P. Resnick, 02/23/1996. (24590 bytes)

"Observations on the use of Components of the Class A Address Space within the Internet", <draft-ietf-cidrd-classa-01.txt>, G.Huston, 12/22/1995. (21347 bytes)

Classless in-addr.arpa delegation", <draft-ietf-cidrd-classless-inaddr-00.txt>, H. Eidnes, G. de Groot, 01/18/1996. (13224 bytes)

"Implications of Various Address Allocation Policies for Internet Routing", <draft-ietf-cidrd-addr-ownership-07.txt>, Y. Rekhter, T. Li, 01/15/1996. (34866 bytes)

"Suggestions for Market-Based Allocation of IP Address Blocks", <draft-ietf-cidrd-blocks-00.txt>, P. Resnick, 02/23/1996. (24590 bytes)

Textbooks

Comer, Douglas E. *Internetworking with TCP/IP Volume 1 Principles, Protocols, and Architecture Second Edition*, Prentice Hall, Inc. Englewood Cliffs, New Jersey, 1991

Huitema, Christian. *Routing in the Internet*, Prentice Hall, Inc. Englewood Cliffs, New Jersey, 1995

Stevens, W. Richard. *TCP/IP Illustrated: Volume 1 The Protocols*, Addison Wesley Publishing Company, Reading MA, 1994

Wright, Gary and W. Richard Stevens. *TCP/IP Illustrated: Volume 2 The Implementation*, Addison Wesley Publishing Company, Reading MA, 1995

Appendix B - Classful IP Addressing

Practice Exercises

1. Complete the following table which provides practice in converting a number from binary notation to decimal format.

Binary	128	64	32	16	8	4	2	1	Decimal
11001100	1	1	0	0	1	1	0	0	128+64+8+4 = 204
10101010									
11100011									
10110011									
00110101									

2. Complete the following table which provides practice in converting a number from decimal notation to binary format.

Decimal	128	64	32	16	8	4	2	1	Binary
48	0	0	1	1	0	0	0	0	$48=32+16=00110000_2$
222									
119									
135									
60									

3. Express 145.32.59.24 in binary format and identify the address class:

   _____

4. Express 200.42.129.16 in binary format and identify the address class:

   _____

5. Express 14.82.19.54 in binary format and identify the address class:

   _____

Solutions to Classful IP Addressing Practice Exercises

1. Complete the following table which provides practice in converting a number from binary notation to decimal format.

Binary	128	64	32	16	8	4	2	1	Decimal
11001100	1	1	0	0	1	1	0	0	**204**
10101010	1	0	1	0	1	0	1	0	**170**
11100011	1	1	1	0	0	0	1	1	**227**
10110011	1	0	1	1	0	0	1	1	**179**
00110101	0	0	1	1	0	1	0	1	**53**

2. Complete the following table which provides practice in converting a number from decimal notation to binary format.

Decimal	128	64	32	16	8	4	2	1	Binary
48	0	0	1	1	0	0	0	0	**0011 0000**
222	1	1	0	1	1	1	1	0	**1101 1110**
119	0	1	1	1	0	1	1	1	**0111 0111**
135	1	0	0	0	0	1	1	1	**1000 0111**
60	0	0	1	1	1	1	0	0	**0011 1100**

3. Express 145.32.59.24 in binary format and identify the classful prefix length.

    10010001.00100000.00111011.00011000   /16 or Class B

4. Express 200.42.129.16 in binary format and identify the classful prefix length.

    11001000.00101010.10000001.00010000   /24 or Class C

5. Express 14.82.19.54 in binary format and identify the classful prefix length.

    00001110.01010010. 00010011.00110110   /8 or Class A