# 1

# THE INTERNET PROTOCOL

## 1.1 HIGHLIGHTS OF INTERNET PROTOCOL HISTORY

The Internet Protocol (IP) has changed everything. In my early days at AT&T Bell Laboratories in the mid-1980s when we used dumb terminals to connect to a mainframe, the field of networking was just beginning to enable the distribution of intelligence from a centralized mainframe to networked servers, routers, and ultimately personal computers. Now that I've dated myself, a little later, many rival networking technologies were competing for enterprise deployments with no clear leader. Deployment of disparate networking protocols and technologies inhibited communications among organizations, until during the 1990s the Internet Protocol, thanks to the widespread embrace of the Internet, became the world's de facto networking protocol.

Today, the Internet Protocol is the most widely deployed network layer* protocol worldwide. Emerging from a U.S. government sponsored networking project for the U.S. Department of Defense begun in the 1960s, the Transmission Control Protocol/Internet

* The network layer refers to layer 3 of the Open Systems Interconnect (OSI) seven-layer protocol model. IP is designed for use with Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) at layer 4, the transport layer, hence the term *TCP/IP protocol suite*. The OSI model and IP networking in general are discussed in the book entitled *Introduction to IP Address Management*. (Ref 11)

*IP Address Management: Principles and Practice*, by Timothy Rooney

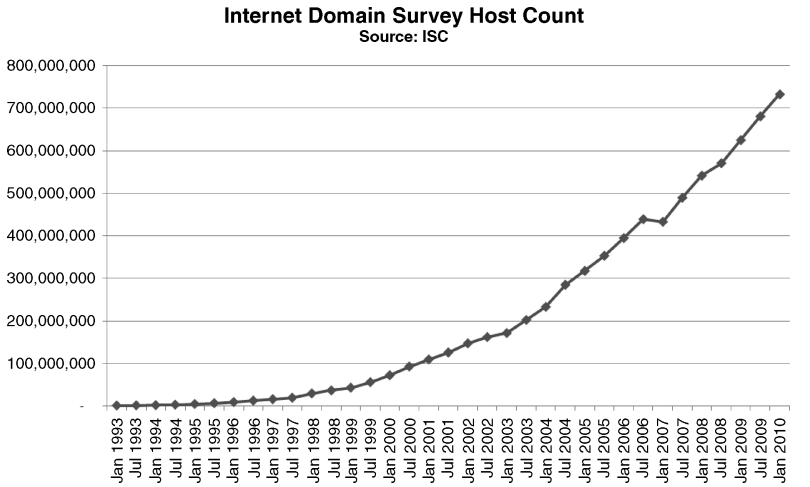


Figure 1.1. Growth of Internet hosts during 1993–2010 (3). *Source:* ISC.

Protocol (TCP/IP) suite has evolved and scaled to support networks from hundreds of computers to hundreds of millions today. In fact, according to Internet Systems Consortium (ISC) surveys, the number of devices or hosts[†] on the Internet exceeded 730 million as of early 2010 with average annual additions of over 75 million hosts *per year* over each of the past 6 years (see Figure 1.1). The fact that the Internet has scaled rather seamlessly from a research project to a network of over 730 million computers is a testament to the vision of its developers and robustness of their underlying technology design.

The Internet Protocol was "initially" defined in 1980 in Request for Comments (RFC[‡]) 760 (1) and 791 (2), edited by the venerable Jon Postel. We quote "initially" because as Mr. Postel pointed out in his preface, RFC 791 is based on six earlier editions of the ARPA (Advanced Research Projects Agency, a U.S. Department of Defense agency) Internet Protocol, though it is referred to in the RFC as version 4 (IPv4). RFC 791 states that the Internet Protocol performs two basic functions: addressing and fragmentation. While this may appear to trivialize the many additional functions and features of the Internet Protocol implemented then and since, it actually highlights the importance of these two major topics for any protocol designer. Fragmentation deals with splitting messages into a number of IP packets so that they can be transmitted over networks that have limited packet size constraints, and reassembly of packets at the destination in the proper order. Addressing is of course one of the key topics of this book, so assuring unique addressability of hosts requiring reachability is critical to basic protocol operation.

[†] The term *host* refers to an end node in the communications path, as opposed to a router or intermediate device. Hosts consist of computers, VoIP telephones, PDAs, and other such IP-addressable devices.

[‡] The Internet Protocol continues to evolve and its specifications are documented in the form of RFCs numbered sequentially. The Internet Engineering Task Force (IETF) is an open community organization with no formal membership and is responsible for publishing RFCs.

The Internet has become an indispensable tool for daily personal and business productivity with such applications as email, social networking, web browsing, wireless access, and voice communications. The Internet has indeed become a key element of modern society. And in case you're interested, the term "Internet" evolved from the lower case form of the term used by the early developers of Internet technology to refer to communications among interconnected networks or "internets."

Today, the capitalized "Internet," the global Internet that we use on a daily basis, has become a massive network of interconnected networks. Getting all of these networks and hosts on them to cooperate and exchange user communications efficiently requires adherence to a set of rules for such communications. This set of rules, this *protocol*, defines the method of identifying each host or endpoint and how to get information from point A to point B over a network. The Internet Protocol specifies such rules for communication using the vehicle of IP packets, each of which is prefixed with an IP header.

### 1.1.1 The IP Header

The IP layer within the TCP/IP protocol suite adds an IP header to the data it receives from the TCP or UDP transport layer. This IP header is analyzed by routers along the path to the final destination to ultimately deliver each IP packet to its final destination, identified by the destination IP address in the header. RFC 791 defined the IP address structure as consisting of 32 bits comprised of a network number followed by a local address. The address is conveyed in the header of every IP packet. Figure 1.2 illustrates the fields of the IP header. Every IP packet contains an IP header, followed by the data contents within the packet, including higher layer protocol control information.

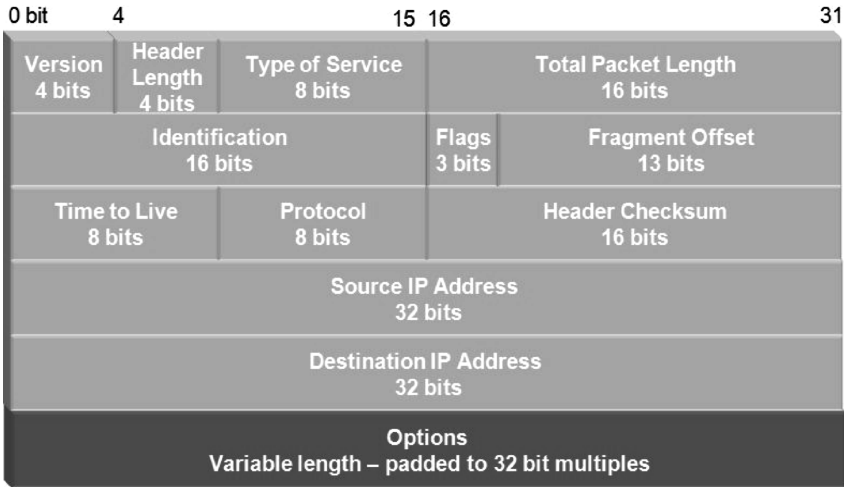


Figure 1.2. IPv4 header fields (1).

*Version*. The Internet Protocol version, 4 in this case.

*Header Length (Internet Header Length, IHL)*. Length of the IP header in 32-bit units called "words." For example, the minimum header length is 5, highlighted in Figure 1.2 as the lightly shaded fields, which consists of 5 words $\times$ 32 bits/word = 160 bits.

*Type of Service*. Parameters related to the packet's quality of service (QoS). Initially defined as ToS (type of service), this field consisted of a 3-bit precedence field to enable specification of the relative importance of a particular packet, and another 3 bits to request low delay, high throughput, or high reliability, respectively.

The original ToS field has been redefined via RFC 2474, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Header" (177). The DS field, or differentiated services field, provides a 6-bit code point (DSCP, differentiated services code point) field with the remaining 2 bits unused. The code point maps to a predefined service, which in turn is associated with a level of service provided by the network. As new code points are defined with respective services treatment by the Internet authorities, IP routers can apply the routing treatment corresponding to the defined code point to apply higher priority handling for latency-sensitive applications, for example.

*Total Length*. Length of the entire IP packet in bytes (octets).

*Identification*. Value given to each packet to facilitate reassembly of packet fragments at the receiving end.

*Flags*. This 3-bit field is defined as follows:

- Bit 0 is reserved and must be 0.
- Bit 1—Don't Fragment—indicates that this packet cannot be fragmented.
- Bit 2—More Fragments—indicates that this packet is a fragment, though this is not the last fragment.

*Fragment Offset*. Identifies the location of this fragment relative to the beginning of the original packet in units of 64-bit "double words."

*Time to Live (TTL)*. A counter decremented upon each routing hop; once the TTL reaches zero, the packet is discarded. This parameter prevents packets from circulating on the Internet forever!

*Protocol*. The upper layer protocol that shall receive this packet after IP processing, for example, TCP or UDP.

*Header Checksum*. A checksum value calculated over the header bits only to verify that the header is not corrupted.

*Source IP Address*. The IP address of the sender of this packet.

*Destination IP Address*. The IP address of the intended recipient of this packet.

*Options*. Optional field containing zero or more optional parameters that enable routing control (source routing), diagnostics (trace route, maximum transmission unit (MTU) discovery), and more.

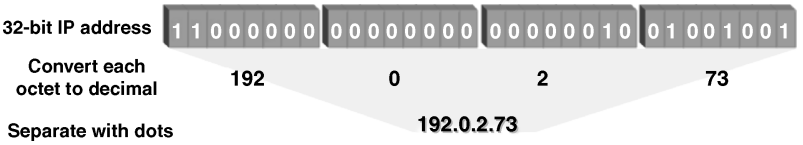


Figure 1.3. Binary to dotted decimal conversion.

It's ok if you find this IP header detail a bit droll. It's only to provide some context, but now let's focus our attention to the source and destination IP address fields and the IP addressing structure.

## 1.2 IP ADDRESSING

The IP address field is comprised of 32 bits. The familiar dotted decimal notation for an IP address reflects the splitting of the 32-bit address into four 8-bit octets. We convert each of the four octets to decimal, and then separate them with decimal points or "dots." This is certainly easier than calculating these 32 bits as one huge number! Consider the 32-bit IP address in Figure 1.3. We simply split this into four octets, convert each octet to decimal, and then separate the decimal representation of each octet by "dots." Hence, the term "dotted decimal."

### 1.2.1 Class-Based Addressing*

RFC 791 (2) defines three classes of addresses: classes A, B, and C. These classes were identified by the initial bits of the 32-bit address as depicted in Figure 1.4. Each class corresponded to a particular fixed size for the network number and local address fields. The local address field could be assigned to individual hosts or further broken down into subnet and host fields, as we'll discuss later.

The division of address space into classes provided a means to easily define different sized networks for different users' needs. At the time, the Internet was comprised of certain U.S. government agencies, universities, and some research institutions. It had not yet blossomed into the de facto worldwide backbone network it is today, so address capacity was seemingly limitless. The other reason for dividing address space into classes on these octet boundaries was for easier implementation of network routing. Routers could identify the length of the network number field simply by examining the first few bits of the destination address. They would then simply look up the network number portion of the entire IP address in their routing table and route each packet accordingly. Computational horsepower in those days was rather limited, so minimizing processing requirements was another consideration. A side benefit of classful addressing was simple readability. Each dotted decimal number represents one octet in binary. As we'll see later when discussing classless addressing, this is not typically the case today.

* Much of the remainder of this chapter leverages material from Chapter 2 of Ref. 11.

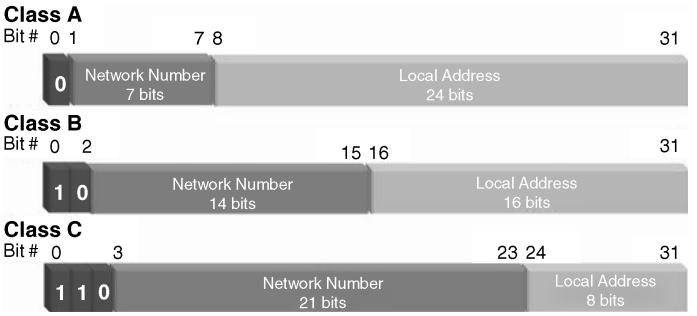


Figure 1.4. Class-based addressing.

Examining this class-based addressing structure, we can observe a few key points:

- Class A networks
  - Class A prefixes begin with binary 0 ($[0]_2$)[†] plus 7 additional bits or 8 network bits total.
  - The network address of all 0s is invalid.[‡]
  - The network address of $[01111111]_2 = 127$ is a reserved address. Address 127.0.0.1 is used for the "loopback address" on an interface.
  - This leaves us with a class A network prefix range of $[00000001]_2$ to $[01111110]_2 = 1$–126 as the first octet.
  - The local address field is 24 bits long. This equates to up to $2^{24} = 16{,}777{,}216$ possible local addresses per network address. Generally, the all 0s local address represents the "network" address and the all 1s is a network broadcast, so we typically subtract these two addresses from our local address capacity in general to arrive at 16,777,214 hosts per class A network. Thus, 10.0.0.0 is the network address of 10.0.0.0/8, and 10.255.255.255 is the broadcast address to all hosts on the 10.0.0.0/8 network.
- Class B networks
  - Class B networks begin with $[10]_2$ plus 14 additional bits or 16 network bits total.
  - The range of class B network prefixes in binary is $[10000000\ 00000000]_2$ to $[10111111\ 11111111]_2$ or networks in the range of 128.0.0.0 to 191.255.0.0, yielding 16,384 network addresses.
  - The local address field is 16 bits long for $65{,}536 - 2 = 65{,}534$ possible hosts per class B network.

[†] To differentiate a binary 0 (1 bit) from a decimal 0 (7–8 bits) in cases where it may be ambiguous, we subscript the number with the appropriate base. Don't worry; we're not digressing into chemistry with discussion of oxygen molecules with the $0_2$ notation, simply "zero base 2."

[‡] Though some protocols such as DHCP use the all 0s address as a placeholder for "this" address.

- Class C networks
  - Class C networks begin with $[110]_2$ plus 21 additional bits or 24 network bits total.
  - The range of class C network prefixes is $[11000000\ 00000000\ 00000000]_2$ to $[11011111\ 11111111\ 11111111]_2$ or networks in the range 192.0.0.0 to 223.255.255.0, yielding 2,097,152 networks.
  - The local address field is 8 bits long for $256 - 2 = 254$ possible hosts per class C network.
- Class D networks (not illustrated in Figure 1.4)
  - Class D networks were defined after RFC 791 and denote multicast addresses, which begin with $[1110]_2$. Multicast is used for streaming applications where multiple users or subscribers receive a set of IP packets from a common source. In other words, multiple hosts having a common multicast address would receive all IP traffic sent to the multicast group or address. There is no network and host portion of the multicast network as members of a multicast group may reside on many different physical networks.
  - The range of class D networks is from $[11100000\ 00000000\ 00000000\ 00000000]_2$ to $[11101111\ 11111111\ 11111111\ 11111111]_2$ or the 224.0.0.0 to 239.255.255.255 range, yielding 268,435,456 multicast addresses.
- Class E networks (not illustrated in Figure 1.4)
  - Networks beginning with $[1111]_2$ (class E) are reserved.

### 1.2.2 Internet Growing Pains

With seemingly limitless IP address capacity, at least as it seemed through the 1980s, class A and B networks were generally allocated to whomever asked. Recipient organizations would then subdivide or subnet* their class A or B networks along octet boundaries within their organizations. Keep in mind that every "network," even within a corporation, needed to have a unique network number or prefix to maintain address uniqueness and maintain route integrity.

*Subnetting* provides routing boundaries for communications and routing protocol updates. Each network over which IP packets traverse requires its own IP network number (network address). As more and more companies sought to participate in the Internet by requesting IP address space, Internet Registries, the organizations responsible for allocating IP address space, were forced to throttle address allocations. Those requesting IP address space from Internet Registries soon faced increasingly stringent application requirements and were granted a fraction of the address space requested. In having to make do with smaller network block allocations, many organizations were forced to subnet on nonoctet boundaries.

Whether on octet boundaries or not, subnetting is facilitated by specifying a *network mask* along with the network address. The network mask is an integer number

*The term *subnet* is frequently used as a verb as in this context, to mean the act of creating a subnet.

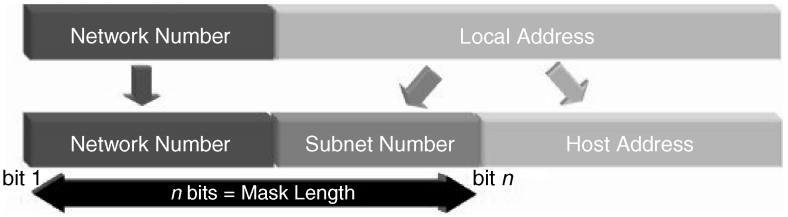


Figure 1.5. Subnetting provides more "networks" with fewer hosts per network.

representing the length in bits of the network prefix. This is sometimes also referred to as the mask length. For example, a class A network has a mask length of 8, a class B of 16, and C of 24. By essentially extending the length of the network number that routers need to examine in each packet, a larger number of networks can be supported, and address space can be allocated more flexibly. This is illustrated in Figure 1.5.

Routers need to be configured with this mask length for each subnet that they serve. This allows them to "mask" the IP address, for example, to expose only the indicated network and subnet bits within the 32-bit IP address to enable efficient routing without relying on address class. Based on this extended network number, the router can route the packet accordingly.

The network address and mask length were originally denoted by specifying the 32-bit mask in dotted decimal notation. This notation is derived by denoting the first $n$ bits of a 32-bit number as 1s and the remaining $32 - n$ bits as 0s, and then converting this to dotted decimal.

For example, to denote a network mask length of 19 bits, you would

- create the 32-bit number with 19 1s and 13 0s: 11111111111111111100000000 00000
- separate into octets: 11111111.11111111.11100000.00000000
- convert to dotted decimal: 255.255.224.0

For example, the notation for network 172.16.168.0 with this 19-bit mask is 172.16.168.0/255.255.224.0.

Thankfully, this approach was superseded by a simpler notation: the mask is now denoted with the network address as <network address>/<mask length>. While the notation is easier to read, it does not save us from the equivalent binary exercise! For example, the 172.16.0.0 class B network would be represented as 172.16.0.0/16. The "slash 16" indicates that the first 16 bits, in this case the first two octets, represent the network prefix.

Here's the binary representation of this network:

| Network Address | *Network Prefix* | Local Address |
|---|---|---|
| `172.16.0.0/16` | *`10101100 00010000`* | `00000000 00000000` |

Let's subnet this network using a 19-bit mask. Expanding this out into binary notation:

| Network Address | *Network Prefix* | ***Subnet*** Local Address |
|---|---|---|
| 172.16.0.0/19 | *10101100 00010000* | ***000***00000 00000000 |
| 172.16.32.0/19 | *10101100 00010000* | ***001***00000 00000000 |
| 172.16.64.0/19 | *10101100 00010000* | ***010***00000 00000000 |
| 172.16.96.0/19 | *10101100 00010000* | ***011***00000 00000000 |
| 172.16.128.0/19 | *10101100 00010000* | ***100***00000 00000000 |
| 172.16.160.0/19 | *10101100 00010000* | ***101***00000 00000000 |
| 172.16.192.0/19 | *10101100 00010000* | ***110***00000 00000000 |
| 172.16.224.0/19 | *10101100 00010000* | ***111***00000 00000000 |

Notice that the class B network bits are depicted under the Network Prefix column in italic font, and we highlighted the subnet bits in larger bold italic font in the Subnet column. Using this 3-bit subnet mask, we effectively extended the network number from 16 bits to 19. By incrementing the binary values of these 3 bits from $[000]_2$ to $[111]_2$ as per the highlighted subnet bits above, we can derive $2^3 = 8$ subnets with this 3-bit subnet mask extension. Routers would then be configured to route using the first 19 bits to identify the network portion of the address by configuring the router serving such a subnet with the corresponding mask length, for example, 172.16.128.0/19, and then having the router communicate reachability to this network via routing protocols. This technique, called variable length subnet masking (VLSM), became increasingly more prevalent in helping to squeeze as much IP address capacity as possible out of the address space assigned within an organization.

The two-layer network/subnet model worked well during the first decades of IP's existence. However, in the early 1990s, demand for IP addresses continued to increase dramatically, with more and more companies desiring IP address space to publish web sites. At the then current rate of usage, the address space was expected to exhaust before the turn of the century! The guiding body of the Internet, the Internet Engineering Task Force, cleverly implemented two key policies to extend the usable life of the IP address space, namely, support of private address space [ultimate RFC 1918 (7)] and classless interdomain routing [CIDR, RFCs 1517–1519 (Ref. 4–6)]. The IETF also began work on a new version of IP with enormous address space during this time, IP version 6, which we'll discuss in the next chapter.

### 1.2.3 Private Address Space

Recall our statement that every "network" within an organization needs to have a unique network number or prefix to maintain address uniqueness and route integrity. As more and more organizations connected to the Internet, the Internet became a potential vehicle for hackers to infiltrate organizations' networks. Many organizations implemented firewalls to filter out IP packets based on specified criteria regarding IP header values, such as source or destination addresses, UDP versus TCP, and others. This guarded

partitioning of IP address space between "internal" and "external" address spaces dovetailed nicely with address conservation efforts within the IETF.

The IETF issued a couple of RFC revisions, resulting in RFC 1918 becoming the standard document that defined the following sets of networks as "private":

- 10.0.0.0—10.255.255.255 (10/8 network)—equivalent to 1 class A.
- 172.16.0.0—172.31.255.255 (172.16/12 network)—equivalent to 16 class B's.
- 192.168.0.0—192.168.255.255 (192.168/16 network)—equivalent to 1 class B or 256 class C's.

The term *private* means that these addresses are not routable on the Internet. However, within an organization, they may be used to route IP traffic on internal networks. Thus, my laptop is assigned a private IP address and I can send emails to my fellow associates, who also have private addresses. My organization in essence has defined a private Internet, sometimes referred to as an intranet. Routers within my organization are configured to route among allocated private IP networks, and the IP traffic among these networks never traverses the Internet.*

Since I'm using a private IP address, someone external to the organization, outside the firewall, cannot reach me directly. Anyone externally sending packets with my private address as the destination address in the IP header will not be able to reach me as these packets will not be routed by Internet routers. But what if I wanted to initiate a connection externally to check on how much money I'm losing in the stock market via the Internet? For employees requiring access to the Internet, firewalls employing network address translation (NAT) functionality are commonly employed to convert an enterprise user's private IP address into a public or routable IP address from the corporation's public address space.

Typical NAT devices provide address pooling features to pool a relatively small number of publicly routable (nonprivate) IP addresses for use on a dynamic basis by a larger number of employees who sporadically access the Internet. The NAT device bridges two IP connections together: the internal-to-NAT device communications utilize private address space, while the NAT device-to-Internet communications use public IP addresses. The NAT device is responsible for keeping track of mapping the internal employee address to the public address used externally.

This is illustrated in Figure 1.6, with the internal network utilizing the 10/8 address space and external or public addressing utilizing the 192.0.2.0/24 space. As per the figure, if my laptop has the IP address 10.1.0.1, I can communicate to my colleague on IP address 10.2.0.2 via the internal IP network. When I access the Internet, my packets need to be routed via the firewall/NAT device in order to map my private 10.1.0.1 address to a public address, for example, 192.0.2.108. The mapping state is maintained in the NAT device and it modifies the IP header to swap out 10.1.0.1 for 192.0.2.108 for outbound packets and the converse for inbound packets.

* Technically, with the use of virtual private networks (VPNs) or tunnels over the Internet, privately addressed traffic may traverse the Internet, but the tunnel endpoints accessing the Internet on both ends do utilize public IP addresses.

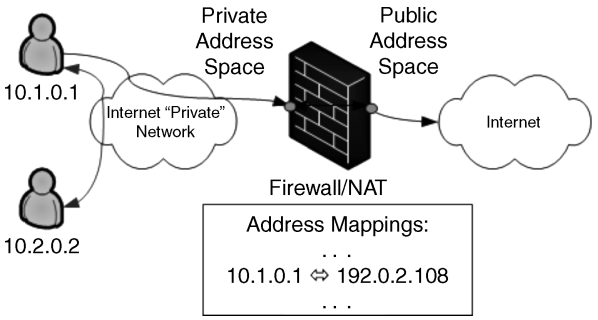


Figure 1.6. Example use of NAT to map private to public addresses.

From an addressing capacity requirements perspective, my organization only needs sufficient IP address space to support these ad hoc internal-to-Internet connections as well as Internet-reachable hosts such as web or email servers. This amount is generally much smaller than requiring IP address space for every internal and external router, server, and host. Implementation of private address space greatly reduced the pressure on address space capacity, as enterprises required far less public address space.

## 1.3 CLASSLESS ADDRESSING

The second strategy put into effect to prolong the life span of IPv4 was the implementation of CIDR, which vastly improved network allocation efficiencies. Like variable length subnet masking, which allows subnetting of a classful network on nonoctet boundaries, CIDR allows the network prefix for the base address block (allocated by a Regional Internet Registry or Internet Service Provider) to be variable. Hence, a contiguous group of four class C's (/24), for example, could be combined and allocated to a service provider as a single /22. This is illustrated in Figure 1.7. If the four contiguous blocks shown, 172.16.168.0/24 to 172.16.171.0/24, are available for allocation, they could be allocated as a single /22, that is, 172.16.168.0/22.

Notice that the darker shaded bits represent the network number, that is, the first 22 bits, which is identical on all four constituent networks. The remaining 10 bits represent

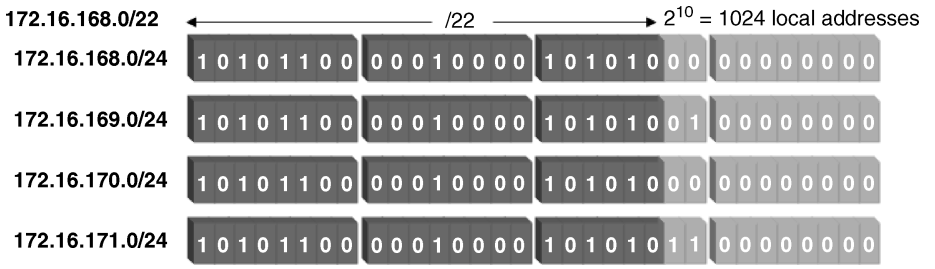


Figure 1.7. CIDR allocation example.

the local address space for host assignment. Since the network address is indicated with all 0s in the local address field, the /22 network is identified as the bit string at the top, namely, 172.16.168.0/22. As you can see, CIDR is very similar to VLSM in terms of the decimal to binary arithmetic required to calculate network addresses on nonoctet boundaries. The extra step of filling in 0s for local addresses outside nonoctet boundary masks introduces an opportunity for error. In addition, VLSM can be applied to a CIDR allocation to further increase the chance of error. But as is usually the case, there's a price to pay for more flexibility. CIDR and VLSM broke down the class walls to provide truly flexible network allocations and subnetworking.

## 1.4 SPECIAL USE ADDRESSES

In addition to private space, certain portions of the IPv4 address space have been set aside for special purposes or documentation. Such IPv4 address allocations include reservations for special use IP addresses, which are summarized below and defined in RFC 3330 (8) and updated in RFC 5735 (9).

| Address Space | Special Use |
|---|---|
| 0.0.0.0/8 | "This" network; 0.0.0.0/32 denotes this host on this network |
| 10.0.0.0/8 | Private IP address space, not routable on the public Internet as per RFC 1918 |
| 127.0.0.0/8 | Assigned for use as the Internet host loopback address, that is, 127.0.0.1/32 |
| 169.254.0.0/16 | The "link local" block used for IPv4 autoconfiguration for communications on a single link |
| 172.16.0.0/12 | Private IP address space, not routable on the public Internet as per RFC 1918 |
| 192.0.0.0/24 | Reserved for IETF protocol assignments |
| 192.0.2.0/24 | Assigned as "Test-Net-1" for use in documentation and sample code |
| 192.88.99.0/24 | Allocated for 6to4 relay anycast addresses (see Chapter 17 for further discussion) |
| 192.168.0.0/16 | Private IP address space, not routable on the public Internet as per RFC 1918 |
| 198.18.0.0/15 | Allocated for use in benchmark tests of network interconnect devices |
| 198.51.100.0/24 | Assigned as "Test-Net-2" for use in documentation and sample code |
| 203.0.113.0/24 | Assigned as "Test-Net-3" for use in documentation and sample code |
| 224.0.0.0/4 | Allocated for IPv4 multicast address assignments (formerly class D space) |
| 240.0.0.0/4 | Reserved for future use (formerly class E space) |
| 255.255.255.255/32 | Limited broadcast on a link |