Introduction to IP Addressing

What are Private IP Addresses?

CIDR and Subnetting

Friday, May 25, 2012

Basic Networking

Classless Inter Domain Routing (CIDR) is a method for assigning IP addresses without using the standard IP address classes like Class A, Class B or Class C. CIDR is a newer addressing scheme for IP Networks which allows for a more efficient allocation of IP addresses than the older method which was by assigning organizations a class of IPs. CIDR was a result of running out of IPv4 addresses as well as addressing the issue with routing tables increasing in size.

There is a maximum number of networks and hosts that can be assigned unique IP addresses using the 32 bit addressing. Traditionally, the Internet assigned "classes" of addresses: Class A, Class B and Class C were the most common. Many large organizations were assigned Class A blocks. Others were assigned Class B blocks. The smaller organizations were assigned Class C blocks. Not all organizations used all of the IP addresses within the block they were assigned. This resulted in an inefficent use of the addressing scheme. For this reason, the Internet was, until the arrival of CIDR, running out of address space very quickly. CIDR effectively solved the problem by providing a new and more flexible way to specify network addresses.

Class	Network Bits	Host Bits	Decimal Range
Class A	8	24	1-126
Class B	16	16	128-191

Class C	24	8	192-223

Using the Class A, B, and C addressing scheme the Internet could support the following 126 Class A networks that could include up to 16,777,214 hosts each. In addition, 65,000 Class B networks that could include up to 65,534 hosts each. Also, you can have over 2 million Class C networks that could include up to 254 hosts each. Of course you have to account that some addresses are reserved for network IDs and broadcast addresses. Because Internet addresses were generally only assigned in these three sizes, there was a lot of wasted IP addresses. For example, if your organization only needed 150 addresses you would be assigned a Class C block. Having a Class C with only using 150 addresses would mean that 104 addresses would be unused. CIDR was developed to be a much more efficient method of assigning addresses.

CIDR to the Rescue

Classless Inter-Domain Routing (CIDR) is a replacement for the old process of assigning Class A, B and C addresses with a generalized network prefix. Instead of being limited to network IDs of 8, 16 or 24 bits, CIDR can specify a range of network prefixes. For example, rather than assigning a Class C block, you can use a network prefix of 27 bits and assign the a block of 32 IP Addresses. This allows for address assignments that can better fit an organization's specific needs with very little waste of IP addresses. A CIDR address includes the standard 32-bit IP address and also information on how many bits are used for the network prefix. For example, in the CIDR address 65.70.30.10/26, the /26 indicates the first 26 bits are used to identify the unique network leaving the remaining bits to identify the hosts.

The CIDR addressing scheme also provides for route aggregation in which a single high-level route entry can represent many specific-level routes in the global routing tables. A high level, backbone network nodes would have network information about a larger portion of the IP scheme, possibly a /13. These large organizations that have been assigned the larger blocks, would redistribute more specific IP blocks to thier customers, such as ISPs. For example, the ISPs that participate on the same network backbone my have a /20 address. Then the ISPs may continue to redistribute IP blocks to their customers that participate on thier network. This process continues and the routes become more and more specific, with very little waste of IP space. In the past, you would get Class A, B, or C address assignments directly from the appropriate Internet Registry such as InterNIC. Under older scenario, you owned the address space. With the introduction of CIDR address assignments and route aggregation, you now rent the address space from your ISP. If you change your ISP, you would need to release the IPs you are currently using and get a new block from your new provider.

Subnetting

CIDR is based on a concept called subnetting. Subnetting allows you to take a class, or block of IP addresses and further chop it up into smaller blocks, or groups of IPs. CIDR and subnetting are virtually the same thing. The term Subnetting is generally used when you use it at the organizational level. CIDR is generally used when you it at the ISP level or higher. In any event, it is only important to know that both terms really refer to the same concept. On local area networks, we use the concept of a subnet mask. A subnet mask is that network prefix we have been discussing so far. For example, if you have a subnet mask of 255.255.255.0, that is the equivalent of a /24 network prefix. If you convert the subnet mask to a binary number, you will note that the first 24 bits are set to one (1). When subnetting an existing block of IPs, you start with the given subnet mask and move it to the right until you have the number of subnets you actually need for your network. For example, say that your provider has given you the following network address: 192.168.1.0/24. Now, you have a requirement to break this network into 6 seperate subnets. If we use a /27 network prefix (subnet mask), we will have a total of 8 subnets at our disposal, each with 32 IPs.

Subnet ID	IP Range	Broadcast IP
192.168.1.0	1-30	.31
192.168.1.32	33-62	.63
192.168.1.64	65-94	.95
192.168.1.96	97-126	.127
192.168.1.128	129-158	.159
192.168.1.160	161-190	.191
192.168.1.192	193-222	.223
192.168.1.224	225-254	.255

In the previous example, our Network ID is 192.168.1.0, with 8 subnet IDs, using a network prefix of /27. Let us take a look at the example in more details. We started with a Network ID of 192.168.1.0/24. The first step is to convert the Network ID to a binary format. This will tell us how many host IDs we have to work with.

```
192.168.1.0     11000000.10101000.00000001.00000000
255.255.255.0   11111111.11111111.11111111.00000000
                --------.--------.--------.--------
AND             11000000.10101000.00000001.00000000
```

So by applying a /24 mask to 192.168.1.0, we have determine that we have 8 bits that we can use for network hosts. These eight bits can be further segmented into separate subnets. We can use the following formula to determine how many hosts we can have for a given subnet. The formula is 2^n-2. In this case n=8, therefore the answer is 254 hosts for a Network ID that has a prefix of /24. If we had 6 bits to work with instead of 8, we could use the same formula: 2^6-2=62 hosts. Keep this formula handy! It can help us determine what our network prefix should be if we needed to subnet our network based on the maximum number of hosts per subnet. You should note that we subtract 2 from the exponent because we cannot assign the first or the last IP in the range to network hosts. The first IP in the range is assigned to the Subnet ID and the last IP in the range is assigned to the broadcast IP.

If you have been given the number of subnets that are required, then you just the same formula: 2^n-2, where n is the number of bits you will use to create your subnets. For example, if you need to create six subnets, you can use three bits, 2^3-2=6. Just as we subtracted 2 when calculating the number of hosts per subnet, we do the same when calculating the total number of subnets for a particular Network ID. Using three bits, provides us with 8 different combinations

```
11000000.10101000.00000001.00000000
11111111.11111111.11111111.11100000
--------.--------.--------.--------

000
001
010
011
100
101
110
111
```

Note: In the past, there were limitations to the use of a subnet with all zeros (000) and all ones (111). Some TCP/IP devices would not allow the use of these subnets. However, with some vendors such

as Cisco Systems, they now support the use of these two subnets when the "ip subnet zero" command is configured.

Now that we know we are using 3 bits for the Subnet ID, we also are aware that we have 5 bits for our hosts. If we use formula shown above: 2^5-2=30, we can calculate that we will have a total of 30 usable IPs for our network hosts. The two IPs that are subtracted are the first IP (Subnet ID) and the last IP (Broadcast IP). Here are our available IP ranges (keep in mind that the first and last subnet may or may not be used depending on whether or not your network devices can support them). Since the /27 produced 8 subnets, but we only required 6 subnets, we can still use this network prefix whether or not our devices support the first and last Subnet ID.

```
SID | Host IDs
000 | 00000-11111 = 0 + up to 31
001 | 00000-11111 = 32 + up to 31
010 | 00000-11111 = 64 + up to 31
011 | 00000-11111 = 96 + up to 31
100 | 00000-11111 = 128 + up to 31
101 | 00000-11111 = 160 + up to 31
110 | 00000-11111 = 192 + up to 31
111 | 00000-11111 = 224 + up to 31
```

Subnet ID	IP Range	Broadcast IP
192.168.1.0 (000)	1-30	.31
192.168.1.32 (001)	33-62	.63
192.168.1.64 (010)	65-94	.95
192.168.1.96 (011)	97-126	.127
192.168.1.128 (100)	129-158	.159
192.168.1.160 (101)	161-190	.191
192.168.1.192 (110)	193-222	.223
192.168.1.224 (111)	225-254	.255

Converting Decimal to Binary

If you are still having trouble with binary numbers, you can use the method described below to help you manually convert the binary to decimal and decimal to binary without the use of a calculator. First, let us look at converting binary to decimal. All you have to do is line up the binary numbers under each column. Each column represents a power of 2. We will convert the binary number 10010101

```
128    64    32    16     8     4     2     1
-----------------------------------------------
  1     0     0     1     0     1     0     1
```

Next, just multiply the decimal in the header with the binary number in the first row. Then add up the results.

```
128    64    32    16     8     4     2     1
-----------------------------------------------
  1     0     0     1     0     1     0     1

128     0     0    16     0     4     0     1
```

Then we add up the results: 128 + 16 + 4 + 1 = 149. To convert a decimal into a binary number, we can use the same matrix, but in reverse. For example, convert the decimal number 181 into binary. The first step is to determine which is the largest number listed in the header that can be divided into 181. The answer is 128.

```
128    64    32    16     8     4     2     1
-----------------------------------------------
  1
```

From there, we subtract 128 from 181. That gives us a difference of 53. The next highest number that divides into 53 is 32. So we place a zero under 64 and a one under 32. If we subtract 32 from 53 we have a difference of 21. Since 16 can divide into 21, we place a one under the 16. We subtract 16

from 21 and we have a difference of 5. Therefore we place a zero under 8, but a one under 4. If we subtract 4 from 5 we are left with 1. Therefore a zero goes under the 2 column, and we are left with one. Since one divides into 1, we place a one under the 1 column. We have no more numbers to work with so the process is complete. You can check your work by reversing the process.

```
128    64    32    16     8     4     2     1
-----------------------------------------------
  1     0     1     1     0     1     0     1
```

The reality is that while this technique is important to learn for anyone that is working with IP addresssing, the reality is that you as a Network administrator will only need to use this technique the few times that you are actually designing subnets. In most cases, if you implement a private IP scheme on your internal network, you will not be too concerned with wasting IP addresses. Most subnets on local area networks do not have more than 254 hosts per subnet so a /24 network prefix is common for subnets that will service network hosts. This technique is more commonly used when you have to inter-connect networks between two routers. For that scenario, you commonly use a /30 (255.255.255.252) network prefix. This gives us two usable IPs, one for each router interface in used in the point-to-point connection.

Recommended Books & Training Resources