

Essential IPAM

OSI, Addressing, Routing, and More

VOLUME 2

Introduction to IP Addressing

This paper examines IP Addressing with an emphasis on understanding how addresses are allocated and managed. It is meant to be an introductory level paper.

Table of Contents

| | |
|--|-----------|
| <i>Section 1 – IP Terminology and Number Format.....</i> | <i>3</i> |
| <i>Section 2 – IPv4 Classful IP Addressing.....</i> | <i>5</i> |
| <i>Section 3 – Classless Internet Domain Routing (CIDR) Addressing</i> | <i>9</i> |
| <i>Section 4 – Device IP Address Configuration.....</i> | <i>14</i> |
| <i>Section 5 – IPv6</i> | <i>16</i> |
| <i>Section 6 – Review.....</i> | <i>17</i> |
| <i>SolarWinds IPAM</i> | <i>18</i> |
| <i>About SolarWinds</i> | <i>19</i> |
| <i>About the Author</i> | <i>19</i> |

Copyright© 1995–2013 SolarWinds. All rights reserved worldwide. No part of this document may be reproduced by any means nor modified, decompiled, disassembled, published or distributed, in whole or in part, or translated to any electronic medium or other means without the written consent of SolarWinds. All right, title and interest in and to the software and documentation are and shall remain the exclusive property of SolarWinds and its licensors. SolarWinds Orion™, SolarWinds Cirrus™, and SolarWinds Toolset™ are trademarks of SolarWinds and SolarWinds.net® and the SolarWinds logo are registered trademarks of SolarWinds All other trademarks contained in this document and in the Software are the property of their respective owners.

SOLARWINDS DISCLAIMS ALL WARRANTIES, CONDITIONS OR OTHER TERMS, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, ON SOFTWARE AND DOCUMENTATION FURNISHED HEREUNDER INCLUDING WITHOUT LIMITATION THE WARRANTIES OF DESIGN, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL SOLARWINDS, ITS SUPPLIERS OR ITS LICENSORS BE LIABLE FOR ANY DAMAGES, WHETHER ARISING IN TORT, CONTRACT OR ANY OTHER LEGAL THEORY EVEN IF SOLARWINDS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

SECTION 1

IP Terminology and Number Formats

Before we dive into the details of what IP addresses are, how they are assigned and how they are routed, we should make sure some of the basic concepts are covered.

For computing purposes, one of three notations is normally used to represent numbers. These are **hexadecimal**, a base 16 system, **decimal**, a base 10 system, and **binary** (base 2) system. Here is a quick review of these number systems and how they are used to represent numbers in IP addressing.

Decimal IP Addresses are base 10 numbers, also known as **dotted (or dot) decimal** format and are in the standard form of XXX.XXX.XXX.XXX, where X is a single digit between 0 and 9 inclusive. 172.16.5.54 is an example of an IP address in dotted decimal format. This format is the most human-readable of the three. We are accustomed to representing numbers base 10.

Binary IP Addresses are binary numbers in the standard format of xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx, where x is either 1 or 0. Each set of eight bits divided by dots is called a **byte** or **octet**. While each individual bit can only be a 1 or a 0, the position of each bit in the octet gives it an **order of significance**. Let's examine one octet and see how this works.

| Bits Position | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---------------|-----|----|----|----|---|---|---|---|
| 1 Value | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 0 Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Each bit in the octet occupies a position relative to the other bits. The bit on the far right is known as the **least significant bit** as its maximum value when the bit is set to 1, is a decimal 1. This bit can only represent a decimal 0 or 1. Moving to the left, the bit in position 2 also has the possible bit values of 1 or 0 and possible decimal values of 0 and 2 only.

So even though each bit can only be a 1 or a 0, the position number the bit occupies allows it to represent a maximum decimal value of 2^n , where n is the bit position. Being that the bit on the far left has the decimal value of 128, it is known as the **most significant bit**.

Let's take a look at the sample dotted decimal IP address of 172.16.5.54 and how it is represented in bit format. Because each section of a dotted decimal address is derived from an octet of bits, they are also referred to as octets. To translate a dotted decimal address into a bit format address, each octet is translated independently. To translate a decimal number to bit format follow these steps:

1. Locate the bit with a decimal value closest to, but less than the decimal value to translate.
2. Set that bit to 1.
3. Subtract the decimal value of that bit from the original decimal number.
4. Locate the bit with a decimal value closest to, but less than the decimal value calculated in step 3.
5. Set that bit to 1.
6. Continue until the sum of the decimal values for all bits set to 1 equals the original decimal number.

So to translate 172 I see that the bit in position 8 has the closest value to 172, so I set that to 1. Now I subtract 128 from 172 to get 44. The bit that is closest to, but less than 44 is in position 6, so I set this to 1 and add its decimal value to 128 to get 160. Getting closer! Now 12 remains. We can't use the bit in position 5 as 16 is larger than 12, but seeing as I need to represent 12, the bits in positions 3 and 4 will take care of this. So the resulting bit format translation of the decimal 172 is 10101100 or:

| | | | | | | | | | | | | | | | | | |
|---------------|-----|---|---|---|----|---|---|---|---|---|---|---|---|---|---|---|-----|
| Bits Position | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | | | | | | | | | |
| Decimal Value | 128 | + | 0 | + | 32 | + | 0 | + | 8 | + | 4 | + | 0 | + | 0 | = | 172 |
| Bit Value | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | | | | | | | | | |

Translating the remaining octets we get the full bit format address of
10101100.00010000.00000101.00110110.

Hexadecimal (Hex) Address Translation is easiest done from bit format rather than directly from a decimal number. Hex IPv6 address numbers are four digits per octet with each digit having a value of 0 to F, making 16 possible values per digit. The standard format of a hex IPv6 address is XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX. Translation of a bit format number to hex is done by breaking each bit octet into two **nibbles**, or half octets and then translating each nibble to a hex value. A decimal to hex value table can be useful in translating to hex as we are more accustomed to dealing with decimal numbers. Below is a bit to decimal to hex table.

| Nibble | Decimal | Hex |
|---------------|----------------|------------|
| 0000 | 0 | 0 |
| 0001 | 1 | 1 |
| 0010 | 2 | 2 |
| 0011 | 3 | 3 |
| 0100 | 4 | 4 |
| 0101 | 5 | 5 |
| 0110 | 6 | 6 |
| 0111 | 7 | 7 |
| 1000 | 8 | 8 |
| 1001 | 9 | 9 |
| 1010 | 10 | A |
| 1011 | 11 | B |
| 1100 | 12 | C |
| 1101 | 13 | D |
| 1110 | 14 | E |
| 1111 | 15 | F |

Below is a table showing each nibble in the IPv4 address 172.16.5.54 and the nibbles hex equivalents.

| | | | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 1010 | 1100 | 0001 | 0000 | 0000 | 0101 | 0011 | 0110 |
| A | C | 1 | 0 | 0 | 5 | 3 | 6 |

So in proper hex format, the dotted decimal address 172.16.5.54 is **0xAC010536** or **AC:01:05:36**. Hex format is most commonly used to represent Media Access Control (MAC) addresses and IPv6 addresses partially due to the ability of hex to represent large numbers in a compact format.

SECTION 2

IPv4 Classful IP Addressing

When IP networking first began, the standard format for any IP address was to specify the network number in the first octet and the remaining three octets were called the *rest*. Only a few governmental and educational networks had the ability to access the Internet, then called the ARPANET (Advanced Research Projects Agency Network). ARPANET began using packet switching protocols like X-25 and switched over to TCP/IP January 1, 1983. It soon became apparent that the 254 network addresses that the original design allowed would be quickly exhausted. RFC 791 was already in the works. This RFC describes in detail several of the internet protocols and a new form of addressing that extends the flexibility of network addressing—Classful IP Addressing. Classful addressing allows for the division of what was the *rest* field to allow for significantly more networks.

Here is how classful address is defined.

- The most significant bits of the first octet signify the network class
- If the most significant bit is 0, this is a class A network and the next 7 bits represent the network number. The remaining 24 bits represent host addresses.
- If the first 2 most significant bits are 1 0, this is a class B network and the remaining 14 bits of the first 2 octets represent the network number and the last 2 octets represent host addresses.
- If the first 3 significant bits are 1 1 0, this is a class C network and the remaining 21 bits of the first 3 octets represent the network and the last octet represents host addresses.
- If the first 3 significant bits are 1 1 1, this is a reserved network range for future use.

Let's take a look at the Classful Subnet Calculator tab in SolarWinds Advanced Subnet Calculator (included in Engineer's Toolset), and see how some classful addresses are divided bit-wise into:

- Class indicators
- Network bits
- Host bits

For these examples, we will use the native subnet masks which are:

- Class A = 255.0.0.0
- Class B = 255.255.0.0
- Class C = 255.255.255.0

Advanced Subnet Calculator

File Edit Tools Skins Help

Export Print Tools Help

solarwinds

Address Details **Classful Subnet Calculator** CIDR Calculator Subnet Addresses

IP Address: 12.15.62.130

Subnet Mask: 255.0.0.0

Mask Bits: 8 Number of Subnets: 1

Host Bits: 24 Hosts per Subnet: 16777214

Subnet Bit Mask: 0nnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh

Generate Subnets

Copy Details

Copy Subnets

Looking at the Subnet Bit Mask section, we can see the assignment of bits for the class indicator, network and host. The first octet of this IP address (21) is represented in bit format as 0010101. Seeing that the most significant bit is a 0, this bit is the class indicator and this is a class A network. The class indicator bits are marked in red in Advanced Subnet Calculator. The next seven bits and blue n's indicating these are network bits and the host bits are marked as green h's.

Below are screen captures showing class B and class C sample network addresses. Note the changes in how bits are shown in the Subnet Bit Mask sections.

IP Address: 135.15.0.0

Subnet Mask: 255.255.0.0

Mask Bits: 16 Number of Subnets: 1

Host Bits: 16 Hosts per Subnet: 65534

Subnet Bit Mask: 10nnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh

Generate Subnets

Class B

IP Address: 215.14.62.0

Subnet Mask: 255.255.255.0

Mask Bits: 24 Number of Subnets: 1

Host Bits: 8 Hosts per Subnet: 254

Subnet Bit Mask: 110nnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh

Generate Subnets

Class C

The classful IP addressing system provides about 3.7 billion unique IP addresses as shown below.

| Class | Applicable Networks | Number of Networks | Number of Address per Network |
|-------|---------------------------|--------------------|-------------------------------|
| A | 1.0.0.0 - 126.0.0.0 | 126 | 16,777,214+ |
| B | 128.0.0.0 - 191.255.0.0 | 16,384+ | 65,534 |
| C | 192.0.1.0 - 223.255.255.0 | 2,097,151 | 254 |

This was assumed to allow for plenty of unique IP addresses for the foreseeable future. It didn't take long for issues to arise with this addressing scheme. These included:

- Wasted addresses. If a small company needed Internet access on 400 devices, a Class C network would be insufficient so these companies were assigned class B networks. This would leave about 65 thousand addresses, or 99.4% of the Class B assigned addresses unused. The division of networks and host boundaries was too rigid.
- Complicated Internet routing tables. With no method of aggregating routes or dividing the Internet into smaller chunks, Internet routers would eventually require over 2 million entries to route to all possible networks.

While classful addressing was the basis for IP internetworking, the limitations became so great that without new methods of controlling IP allocation, the available IP address space would not have lasted into the late 90's. It was clear that steps had to be taken to preserve IPv4 address space wherever possible.

Private IP Addresses

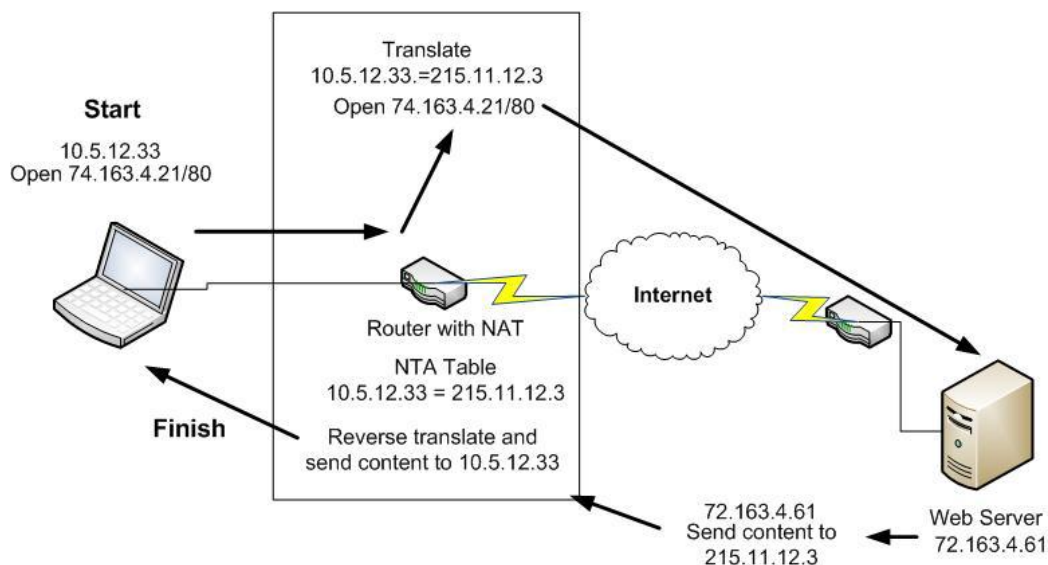
To help alleviate the usage of registered IP addresses for systems that did not require direct connection to the Internet or other registered IP networks, a set of address was set aside. These addresses can be used for any system that communicates within a private network. Because these addresses are never allowed on the Internet, they can be reused by any number of private networks. Below is a table of the private IP address ranges as described by RFC 1918.

| Address Space | Networks |
|-------------------------------|----------------------|
| 10.0.0.0 – 10.255.255.255 | 1 Class A Network |
| 172.13.0.0 – 172.31.255.255 | 16 Class B Networks |
| 192.168.0.0 – 192.168.255.255 | 256 Class C Networks |

Private IP addressing has undoubtedly saved the unnecessary waste of assigning registered addresses to every IP enabled device, and has also helped reducing hijacking (unauthorized use) of registered address space. There is one severe limitation to private IP addressing—the IP private addresses used make an island network, unable to communicate with outside networks. This is where Network Address Translation (NAT) comes in.

Network Address Translation (NAT)

Private IP addresses do not help very much if systems with private addresses cannot access services outside the private network space. Here is how NAT works to solve this issue:



- A system with the private IP address 10.5.12.33. needs to access a Web server on the Internet with the registered IP address 72.163.4.161.

- Because private IP addresses like 10.5.12.33. are not allowed on the Internet, this address must be translated to one that is Internet routable.
- The Internet router connected to the 10 net has NAT enabled and so translates the 10.5.12.33 address to a registered IP address from its list of configured, registered addresses, such as 215.11.12.3.
- The NAT router then makes a request to open the website at 72.163.4.61 from the translated address of 215.11.12.3.
- The Web server returns the Web content to the NAT router for delivery to 215.11.12.3.
- The NAT router reverse translates the target IP Web response to the original IP address of 10.5.12.33.

In this example, the NAT router only has one translation to keep track of. Typically, NAT routers have from scores to hundreds of translations to maintain. The NAT router keeps these translations in one of two types:

- Static translations, used where a private internal system always connects to the same public system.
- Dynamic translation, used for allowing private systems to connect to various public systems.

Dynamic systems allow a relatively small pool of public addresses to be used for a large population of users on a private network. When a private user on a dynamically NAT'ed network requests access to a public addressed system, the NAT server looks for the next available registered address in its NAT pool and maps the original private address of the requestor to an available public address. Once the connection to the public device is no longer needed, the NAT router releases the registered address back into the NAT pool for reuse.

An NAT router may multiplex a single registered IP address by translating the layer 4 port number as well as the private IP address. This type of translation is called Port Address Translation (PAT). Here is a simplified, hypothetical PAT table. Because the registered IP address is using unique ports for each internal address and port, the 215.11.12.3 registered address can be used simultaneously in these multiple PAT devices avoid translating to well-known ports as listed in the INNA Well Known Ports.

| Inside Address | Inside Port | Registered Address | Outside Port |
|----------------|-------------|--------------------|--------------|
| 10.5.21.31 | 216 | 215.11.12.3 | 50432 |
| 10.5.21.66 | 56 | 215.11.12.3 | 63123 |
| 10.5.21.39 | 6784 | 215.11.12.3 | 49732 |
| 10.5.21.21 | 2151 | 215.11.12.3 | 55351 |

One issue with NAT is the UDP and TCP packets contain header checksums that are calculated based upon the TCP/UDP/IP header. If an NAT system simply changes an IP address or port number, the checksum will now be in error when recalculated by the end system. Therefore, the NAT/PAT system must recalculate the checksum fields before forwarding packets. The same applies when the NAT/PAT system reverse-NAT's the packets and sends to the original, privately addressed system.

SECTION 3

Classless Internet Domain Routing (CIDR) Addressing

In Section 2 - Classful Addressing, we saw that the most significant bits of the first octet in an IP address determined the class of that address. These are the class indicator bits and all remaining bits in a classful address are either network or host bits. In CIDR addressing the bits can only represent network field bits or host field bits. In classful addressing, the network address value could only fall on a byte boundary. In CIDR addressing, any number of the most significant bits can represent the network number and the remaining bits represent host addresses. Because there is no longer a distinction of network class based on the most significant bits, this system is called classless.

The magic behind CIDR is the use of a Variable Length Subnet Mask (VLSM). VLSM allow for the network/host address boundary to occur anywhere in the 32 bit IP address. A network sample address in CIDR notation is 215.15.62.0/24. The /24 indicates that the first 24 bits are network bits and the remaining bits are host bits. Let's compare the classful class C network we saw in Advanced Subnet Calculator with the same network number in CIDR notation.

IP Address: 215.14.62.0
Subnet Mask: 255.255.255.0
Mask Bits: 24
Host Bits: 8
Number of Subnets: 1
Hosts per Subnet: 254
Subnet Bit Mask: 110nnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh

Classful 215 Network

Address Block: 215.14.62.0 /24
CIDR Mask: 255.255.255.0 or 24 bits
Subnet Mask: 255.255.255.0
Mask Bits: 24
Host Bits: 8
Number of Subnets: 1
Hosts per Subnet: 254
Subnet Bit Mask: nnnnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh

CIDR 215.15.62.0/24 Block

Note how *all* of the leading 24 bits are network bits. Now we'll move the network bit boundary by changing the CIDR prefix size to 26 bits (/26).

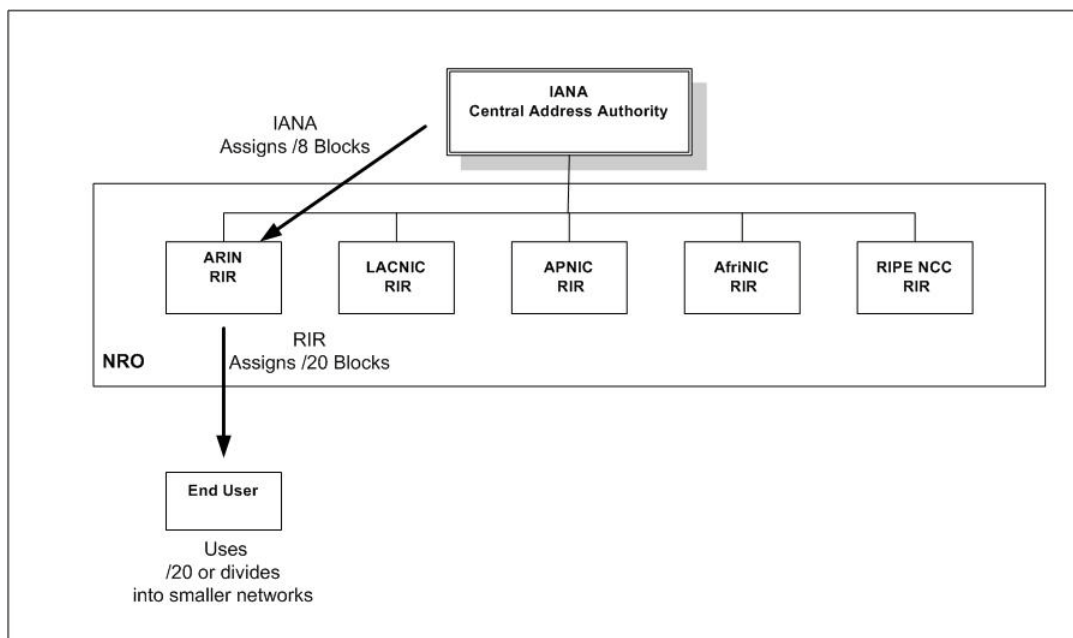
Address Block: 215.15.62.0 /26
CIDR Mask: 255.255.255.192 or 26 bits
Subnet Mask: 255.255.255.192
Mask Bits: 26
Host Bits: 6
Number of Subnets: 1
Hosts per Subnet: 62
Subnet Bit Mask: nnnnnnnn.nnnnnnnn.nnnnnnnn.nnhhhhhh

CIDR 215.15.62.0/26 Block

Now all of the first 26 bits are network and 6 bits are left for host addressing. This flexibility makes it possible to assign blocks of registered IP addresses to better fit the actual need of each requesting organization. Now we'll take a look at how these requests are made, and why you would want to use CIDR subnetting and supernetting.

Requesting a Registered CIDR Block

Regional Internet Registries (RIRs) are organizations responsible for managing the requests for CIDR blocks from within their individual regions. These RIRs participate together in an entity called the Number Resource Organization (NRO). The NRO assists RIRs with the coordination of CIDR blocks. All of the RIRs are assigned their CIDR blocks, (usually in /8 blocks) from Internet Assigned Numbers Authority (IANA), the big daddy of CIDR assignment. Here is how this all typically works:



Each RIR is responsible for assigning address blocks of the proper size to requestors in their region. The RIRs and their respective regions are:

- ARIN – American Registry for Internet Numbers. North America, some Caribbean and Central American nations and Antarctica.
- LACNIC – Latin American and Caribbean Registration Authority. Central and South America as well as Caribbean nations.
- APNIC - Asia-Pacific Network Information Centre. All Asia-Pac nations.
- AfriNIC - African Network Information Center. All of Africa
- RIPE NCC – Reseaux IP Europeens Network Coordination Centre. Europe, Central Asia and the Middle East.

The RIRs take the /8 blocks assigned to them and break them up into smaller networks by shifting the CIDR bits. Below is a sample from the IANA Address Registry taken at the time this paper was written.

| | | | | | |
|-------|----------|---------|----------------|-------------|--|
| 095/8 | RIPE NCC | 2007-07 | whois.ripe.net | ALLOCATED | |
| 096/8 | ARIN | 2006-10 | whois.arin.net | ALLOCATED | |
| 097/8 | ARIN | 2006-10 | whois.arin.net | ALLOCATED | |
| 098/8 | ARIN | 2006-10 | whois.arin.net | ALLOCATED | |
| 099/8 | ARIN | 2006-10 | whois.arin.net | ALLOCATED | |
| 100/8 | IANA | | | UNALLOCATED | |
| 101/8 | IANA | | | UNALLOCATED | |

The left numbers represent abbreviated CIDR blocks. The 099/8 block is the CIRD network block 99.0.0.0/8. As is, this block is one single network with 2^{24} host addresses, just like the case of a classful, class A network. The second column lists the RIR the block has been assigned to. Here we can see that 095/8 was assigned to RIPE NCC and 099/8 has been assigned to ARIN. 100/8 has not yet been assigned to an RIR, thus IANA is listed as the address authority. The date range listed is the range when addresses from that block were assigned to a RIR and the RIR was actively working on further assigning. The *ALLOCATED* note does not mean that all of the addresses in this block are used, it just indicates that the IANA has allocated the block to an RIR.

It's the job of each RIR to distribute these /8 blocks in smaller blocks to preserve and control IPv4 and IPv6 address space. As mention earlier, an RIR will normally assign a /20 block to a large requestor, and smaller blocks to smaller requestors. The trick is splitting the /8 into the smaller /20 blocks. Again, we'll look at Advanced Subnet Calculator to see how this is accomplished.

Address Block
100.0.0.0
/8

CIDR Mask
255.0.0.0
or
8
bits

Subnet Mask
255.255.240.0

Generate Subnets

Mask Bits
20
Number of Subnets
4096

Host Bits
12
Hosts per Subnet
4094

Subnet Bit Mask
nnnnnnnn.ssssssss.sssshhhh.hhhhhhhh

Copy Details

Copy Subnets

| Subnet | Mask | Inverse Mask | Subnet Size | Host Range | Broadcast |
|------------|---------------|--------------|-------------|----------------------------|--------------|
| 100.0.0.0 | 255.255.240.0 | 0.0.15.255 | 4094 | 100.0.0.1 to 100.0.15.254 | 100.0.15.255 |
| 100.0.16.0 | 255.255.240.0 | 0.0.15.255 | 4094 | 100.0.16.1 to 100.0.31.254 | 100.0.31.255 |
| 100.0.32.0 | 255.255.240.0 | 0.0.15.255 | 4094 | 100.0.32.1 to 100.0.47.254 | 100.0.47.255 |
| 100.0.48.0 | 255.255.240.0 | 0.0.15.255 | 4094 | 100.0.48.1 to 100.0.63.254 | 100.0.63.255 |

Here is what is happening. The 100/8 block was assigned a 20 bit subnet mask, thus changing 12 of the bits that were previously host bits to subnet bits. The term *subnet* is shorthand for *subdivided network*. In other words, a large network divided into smaller ones. *Subnetwork* is really just a special term for a network. So we can take all the *s* bits above assigned to subnetworks and just consider them *n* or

network bits. All of the subnetworks listed in the lower section of the above screen then are really unique /20 CIDR blocks and can be assigned to requestors as registered /20 networks. As you can see above, each /8 block that the RIR receives can distributed as 4096 registered /20 blocks. Even more networks can be registered when the RIR issues networks with more network bits, such as /22.

Once a requester (usually an ISP) receives one of these /20 blocks, for example the 100.0.16.0/22 block, the requester is free to further divide up this block as needed. Let's say the ISP needs to allocate address blocks to individual customers. While it is certainly possible to do a bitwise analysis to determine the subnet addresses, host addresses and broadcast addresses, this is a bit like calculating sine and cosine factors by hand. We'll make life a bit more enjoyable and use the Advanced Subnet Calculator again.


First let's take a look at the /20 block with no subnet applied.

| Subnet | Mask | Inverse Mask | Subnet Size | Host Range | Broadcast |
|------------|---------------|--------------|-------------|----------------------------|--------------|
| 100.0.16.0 | 255.255.240.0 | 0.0.15.255 | 4094 | 100.0.16.1 to 100.0.31.254 | 100.0.31.255 |

The subnet calculator shows us that there are 4096 addresses available, minus 100.0.16.0 for the network address and 100.0.31.255 for broadcast. Now, by altering the subnet mask bits we can create subnets of the /20 block to fit each need. This ISP has a pool of customers requiring up to six registered addresses with a projected growth of eight additional addresses. So looking at the subnet calculator again, the ISP will move the subnet mask bits to create networks of at least fourteen useable addresses.

Address Block /20


CIDR Mask or bits


Subnet Mask  Generate Subnets

Mask Bits Number of Subnets

Host Bits Hosts per Subnet

Subnet Bit Mask

 Copy Details

 Copy Subnets

| Subnet | Mask | Inverse Mask | Subnet Size | Host Range | Broadcast |
|--------------|-----------------|--------------|-------------|------------------------------|--------------|
| 100.0.16.0 | 255.255.255.240 | 0.0.0.15 | 14 | 100.0.16.1 to 100.0.16.14 | 100.0.16.15 |
| 100.0.16.16 | 255.255.255.240 | 0.0.0.15 | 14 | 100.0.16.17 to 100.0.16.30 | 100.0.16.31 |
| 100.0.16.32 | 255.255.255.240 | 0.0.0.15 | 14 | 100.0.16.33 to 100.0.16.46 | 100.0.16.47 |
| 100.0.16.48 | 255.255.255.240 | 0.0.0.15 | 14 | 100.0.16.49 to 100.0.16.62 | 100.0.16.63 |
| 100.0.16.64 | 255.255.255.240 | 0.0.0.15 | 14 | 100.0.16.65 to 100.0.16.78 | 100.0.16.79 |
| 100.0.16.80 | 255.255.255.240 | 0.0.0.15 | 14 | 100.0.16.81 to 100.0.16.94 | 100.0.16.95 |
| 100.0.16.96 | 255.255.255.240 | 0.0.0.15 | 14 | 100.0.16.97 to 100.0.16.110 | 100.0.16.111 |
| 100.0.16.112 | 255.255.255.240 | 0.0.0.15 | 14 | 100.0.16.113 to 100.0.16.126 | 100.0.16.127 |


Now the ISP has 265 individual networks, each containing fourteen registered addresses it can allocate to customers. The ISP would give the customer two important pieces of information at this point.

1. The IP address of the Internet side of the company's gateway router.
2. The CIDR block for use of other systems requiring access to the Internet. These might be machines such as NAT/PAT routers or DMZ servers.

The end customer can further subnet the assigned block as shown below.

Address Block /28


CIDR Mask or bits


Subnet Mask  Generate Subnets

Mask Bits Number of Subnets

Host Bits Hosts per Subnet

Subnet Bit Mask

 Copy Details

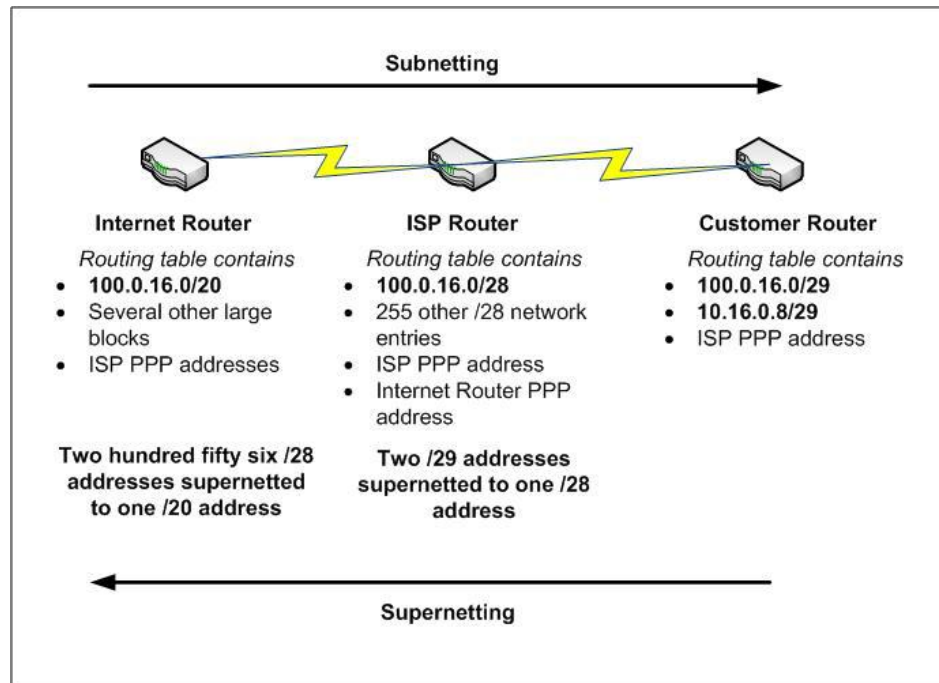
 Copy Subnets

| Subnet | Mask | Inverse Mask | Subnet Size | Host Range | Broadcast |
|------------|-----------------|--------------|-------------|---------------------------|-------------|
| 100.0.16.0 | 255.255.255.248 | 0.0.0.7 | 6 | 100.0.16.1 to 100.0.16.6 | 100.0.16.7 |
| 100.0.16.8 | 255.255.255.248 | 0.0.0.7 | 6 | 100.0.16.9 to 100.0.16.14 | 100.0.16.15 |

It is possible to subnet to as small as a /31 (per RFC 3021) block for PPP connections where only two systems are interconnected by that subnet.

So what then is this supernetting thing? Let's take the /8 address block we submitted and look at how the end user addresses can be supernetted to help keep Internet routing tables manageable.

For the IP subnet 100.0.16.8/29, here is how this subnet is reached using supernetting, the opposite of subnetting.



SECTION 4

Device IP Address Configuration

In the above section we examined IP addresses at the network level. Of course for any individual system to connect to these networks the system needs certain IP information configured. Here is a sample of the minimum IP configuration of a Windows machine.

```
H:\>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : tul.solarwinds.net
    IP Address. . . . . : 10.110.62.130
    Subnet Mask . . . . . : 255.255.254.0
    Default Gateway . . . . . : 10.110.62.1
```

This information specifies:

- The DNS domain
- The machine's unique IP address
- The subnet mask
- The default gateway, where this machine should send all IP communications

While it is possible (and sometimes correct) to manually set these parameters using *static addressing*, most of the time IP addresses are automatically assigned.

Dynamic Host Configuration Protocol (DHCP)

Usually, a device receives its IP configuration from a DHCP server by making a DHCP request broadcast on UDP port 67. The DHCP server will then respond with addressing data on UDP port 68. Below is a packet capture of a Windows machine making a DHCP request for an IP address.

| No. | Time | Source | Destination | Protocol | Info |
|-----|-----------|-------------|-----------------|----------|--|
| 147 | 16.838522 | 0.0.0.0 | 255.255.255.255 | DHCP | DHCP Request - Transaction ID 0x1f640b88 |
| 148 | 16.847302 | 10.110.62.2 | 10.110.62.130 | DHCP | DHCP ACK - Transaction ID 0x1f640b88 |
| 149 | 16.852702 | 10.110.62.3 | 10.110.62.130 | DHCP | DHCP ACK - Transaction ID 0x1f640b88 |

| |
|---|
| Frame 147 (383 bytes on wire, 383 bytes captured) |
| Ethernet II, Src: dell_0d:2b:74 (00:1c:23:0d:2b:74), Dst: Broadcast (ff:ff:ff:ff:ff:ff) |
| Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255) |
| User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67) |
| Source port: bootpc (68) |
| Destination port: bootps (67) |
| Length: 349 |
| Checksum: 0x3f5c [correct] |
| Bootstrap Protocol |

Here is what is happening in this address request.

1. In packet number 147, the host without an IP address (0.0.0.0) makes a broadcast (255.255.255.255) request for a DHCP assigned address.
2. In packet number 148, the DHCP server at 10.110.62.3 offers the machine previously at 0.0.0.0 the IP address 10.110.62.130.
3. In packet number 149, the host now using 10.110.62.30 acknowledges the assignment of the IP address, gateway address, and any other addressing configuration sent from the server.

If the requesting device fails to get a response from a DHCP server, it may be because the device is on a very small network with devices configured to allow IP auto configuration. If this is the case, the device will configure an IP address from a reserved *link-local* block (169.254.0.0/16). This is a non-routable address block for use on island auto configuring IP network segments only. It is for plug-and-play IP segments like interconnecting game consoles.

There are several options in DHCP implementations to allow for features such as the maximum time a device may use an IP (maximum lease time), the request to reuse the last address the machine had been assigned (sticky addressing), static assignments, and the control of address pool ranges. Enterprise level DHCP servers are very common and are even embedded into devices such as routers. For carrier class DHCP, the server requirements are much more demanding. The reason for this is twofold. First, a large ISP or cable provider needs to track very large numbers of addresses. Secondly, a catastrophic event such as a metropolitan power outage could cause many hundreds of thousands of devices to all make DHCP requests at once when power returns.

More complex devices, such as routers, require multiple IP addresses for physical interfaces, subinterfaces and loopbacks. Assignment of addresses to these devices is made according to the internetworking functions they will perform and the IP addressing and routing designs they will support.

SECTION 5

IPv6

You've probably noticed the repeating theme about the eventual exhaustion of available IPv4 addresses. While the use of CIDR, NAT, and private addressing has helped to conserve the IPv4 address pool, address exhaustion is inevitable. Most sources now estimate the time to IPv4 address exhaustion in months. While there are several features of IPv6, the main impetus for development and implementation is the extension of the IP address space. IPv6 expands the address space from the 32 bits in IPv4 to 128 bits. This not only allows for allocation of more addresses, but also allows for better planning of address allocation and considerations to improve supernetting abilities. IPv6 has been designed from the ground up to avoid many of the shortcomings of IPv4.

The 128 bit address field allows for 2^{128} unique addresses. To give you an idea of how large this is, if this number were divided evenly by everyone alive today, we would each receive about $1,000,000,000^3$ (one billion cubed) addresses. This is actually a bit of an exaggeration since the half of the 128 bit address field is typically used for subnet identification. Still, this leaves us with a huge number of addresses. The 128 bit address field in an IPv6 packet cannot be accommodated by an IPv4 packet header so a new IPv6 packet format was created. Because of the differences in packet formats, the two versions are not directly interoperable. This does not mean that they can't coexist on the same network; it means that they will operate independently.

Address Formats

The IPv6 standard address format is written in eight blocks of four hex digits as shown below.

2001:24C8:85D3:08DE:3145:C82E:0371:1237

CIDR notation is standard for IPv6 so the network address is **2001:24c8:85d3:08de/64**. IPv6 addresses are controlled by the IANA and RIRs but allocated differently due to the size difference. Here is how the above address is broken down by addressing authority and therefore route aggregation.

2001: 24 C8:85D3: 08DE: 3145:C82E:0371:1237

- **2001:** is the Top Level network Aggregator (TLA) administered by the IANA. The most significant three bits are reserved and the thirteen remaining bits create 8,192 TLA networks which are allocated to RIRs who then allocate them to large scale ISPs.
- The next 16 bits (**24**) are reserved for possible expansion of the fields directly to the right or left of these bits.
- **C8:85D3:** is the Next Level Aggregator (NLA) field. The ISPs use this field to subnet TLAs and assign to lower level ISPs or end customers.
- **08DE:** provides 16 bits for the end customer to subnet the NLA.
- **3145:C82E:0371:1237** is the host address.

Address abbreviation is allowed for hex address blocks with leading zeros or entire hex blocks that are set to zero. For example the address

2001:24C8:85D3:00DE:0145:0000:0000:1237

could be abbreviated as

2001:24C8:85D3:DE:145::1237.

Leading zeros are simply dropped and groups of zero hex blocks are replaced by a double colon. CIDR notation is always used to designate any subnetting. The above address has a /64 network designated as

2001:24C8:85D3:DE::/64. A 48 bit subnet of this is **2001:24C8:85D3::/48**. Because IPv4 URL

addressing uses the **http://IP.IP.IP.IP:Port/** format to represent an IP address and port, problems can arise when using colon separated IPv6 address and ports within a URL. To avoid this IPv6 URLs require bracketing of the IP address portion of the URL. For example: **https://[2001:24C8:85D3:DE:145::1237]:1415/**.

Special Address Types

- Local loop-back - `::1/128`. Same as `127.0.0.0` in IPv4.
- Link-local – `FE80::/10` block is reserved for these auto configuring, non-routable addresses.
- Unique local – `FC00::/7`. Same as IPv4 private addresses.
- Solicited Multicast Address - `FF02::1:FF00:0/104` used for layer 2 address resolution.
- Default route - `::/0`. Same as IPv4 `0.0.0.0/0.0.0.0`

Other Notable IPv6 Features

- Mandatory IPsec creating secure links between Internet devices.
- Jumbograms allowing for single packets with a payload of up to 4GB.
- Hierarchical multi-cast allowing for defined multi-cast to local, subnet, supernet or global levels.
- IPv4 tunneling allowing IPv6 to be repackaged for transport over IPv4 only segments. Obviously a temporary patch.

IPv6 adoption is gaining momentum now that the services required to operate in an IPv6 environment, such as IPv6 DNS, are becoming available. ISP's are now in the process of implementing IPv6 on their productions networks. Because of the lack of global adoption, systems using IPv6 are usually *dual stacked*, running both IPv4 and IPv6 protocol stacks.

SECTION 6

Review

- IP networking has been in use since the early seventies. At that time several other network protocols were in use. With the mass adoption of the internet, IP became the protocol of choice as it is used to route traffic on the Internet.
- Originally IP used only the first octet for network addressing. Later classful addressing was introduced to allow for controlled and structured use of the IPv4 address space.
- New technologies such as private network addressing, NAT/PAT, and CIDR/VSLM were created to further help preserve the remaining available address space.
- Internet routable (public) IP addresses are assigned by the IANA to one of five RIRs, then from the RIR to the requestor, typically an ISP.
 - Each one of these entities subnets the assigned space for most efficient use.
- From the Internet backbone to the end network, networks are subnetted. Going the other direction, subnets are aggregated into supernet addresses for more manageable global routing.
- End user devices are typically addresses using DHCP with leased IP addresses. Internetworking devices are typically addressed manually with static IP addresses.
- IPv6 has been created to eventually replace IPv4 and allow for a much larger pool of usable IP addresses.
 - IPv4 uses 32 bit addressing
 - IPv6 uses 128 bit addressing

Hope this is helpful and I'll see you on [thwack](#)!

SolarWinds IP Address Manager

Are you still using spreadsheets to manage your IP space? SolarWinds IP Address Manager (IPAM) enables you and your team to ditch your spreadsheets and switch to easy-to-use, centralized IP address management software. Now it's easier than ever to manage and monitor Microsoft® DHCP and DNS, as well as Cisco® DHCP servers, all from a single, intuitive Web console.

With SolarWinds IPAM, you can:

- Centrally manage, monitor, alert, & report on entire IP infrastructure
- Maintain Microsoft® DHCP/DNS & Cisco® DHCP services from a single Web interface
- Optimize IP space utilization & avoid IP conflicts via automatic scans & preventative alerts
- Deliver role-based access control along with detailed event recording & activity logs
- Gain critical insight into IP address space through real-time views & historical tracking

[Get more information on IP Address Manager here.](#)

[Evaluate IP Address Manager in a live demo environment here.](#)

[Download the fully functioning, 30-day evaluation here.](#)

About SolarWinds

Founded in 1999, SolarWinds delivers powerful and affordable IT management and monitoring software to over 100,000 customers worldwide – from Global 1000 enterprises to small businesses. Named by Forbes as one of the top 10 fastest growing technology companies, SolarWinds is improving the way IT management software is developed, priced, purchased, delivered, and used.

Our IT monitoring and management software is built for SysAdmins and network engineers who need powerful, affordable, and easy-to-use solutions that save time and simply get the job done. Evaluating, buying, deploying, and using enterprise software shouldn't be complex. IT management solutions from SolarWinds are easy to try, buy, deploy and use. That's unexpected simplicity.

SolarWinds is headquartered in Austin, Texas, with sales and product development offices around the world. Join our online community of experts at thwack.com!

About the Author

Andy McBride is a Technical Specialist for SolarWinds focusing on making knowledge of networking and network management accessible to customers and prospects of all levels. The “Essential IPAM” series is specifically written for an audience with limited prior exposure to these technologies. His technical background includes seven years at International Network Services (INS) as a Network Engineer and Managing Consultant, three years as a Novell Certified Instructor and five years as a Network Performance Products Manager with BT-Infonet. Prior to entering technology, Andy worked in aerospace on projects such as the SR-71, F-117, F-22, L-1011, F-18, and the space shuttle main engine and has a degree in Chemistry from California State University Northridge. Go Matadors!