In this blog post, I’m going to cover:

* Explanation of IPv6 in terms and terminology for those of you familiar with IPv4
* How you officially get a block of IPv6 address
* What the equivalent of Private (internal) network addressing is in IPv6
* Understand IPv6 addressing
* How to subnet IPv6
* How the concept of gateways and routing works in IPv6

This is the first of many technical blog posts I’m going to post on IPv6 architecture and implementation for a Microsoft Windows-based environment.  I started off with a basic introduction of the IPv4 problem and covered a handful of FAQs in my initial post <http://www.networkworld.com/community/blog/getting-serious-ipv6-windows-networking-envir>, now on to the “How to” guides…

**Explanation of IPv6 in terms and terminology for those of you familiar with IPv4**

IPv6 is similar in many ways to IPv4 addressing, basically every device has to have an IP address, there is name resolution of IPv6 addresses to common names, dynamic addressing, static addressing, routing, etc.  However when drawing up the specification for IPv6, rather than doing things “exactly” like IPv4 (good and bad), IPv6 improved upon a handful of things (that I’ll explain here) to simplify addressing, routing, improve security, and improve performance and efficiency of IPv6 communications compared to IPv6.  So as much as IPv6 addresses are really long and you might assume that would put a huge increase in traffic and payload on IPv6 over IPv4, what was done “inside” IPv6 actually makes it more efficient in many ways.

So the following are terminology in IPv4 terms and how they are addressed in IPv6:

* IP Address:  Each device will have an IP address still, but instead of an IPv4 address, it’ll have an IPv6 address.  Other than the length and slightly different look, this concept is identical
* Subnet Mask:  We used to do subnet masks in IPv4 with notation like 255.255.255.0, but in IPv6, while we still do subnetting, the notation is different in two ways.  We now write subnets using a slash and a number that denotes the masking.  So it’ll look like IPV6ADDRESS/64 or IPV6ADDRESS/112.  But when you actually key in the IPv6 address on a system, that /64 or /112 will convert to a hexadecimal number that’ll be in the middle of the IPv6 Address.  So when you see an IPv6 address, while it is really long, it actually includes the Network Address: Subnet: Device IP Address in that long address string.  More on this in the “Understand IPv6 Addressing” section below
* Gateway Address:  The concept of the network gateway in IPv6 is the same as in IPv4, a gateway address will be designated noting how traffic can be routed out of the current subnet (technically the IPv6 Gateway address is not a formalized standard in IPv6, however Microsoft has included a Gateway setting in their IP Configuration properties page)

So all of the concepts remain the same, but you’ll see when we get to the IPv6 addressing section that the long IPv6 address includes the Network Address, Subnet, and Unique Device Address all togheter

**How you officially get a block of IPv6 address**

So the next question everyone always asks is “How do I get an official IPv6 block of addresses?”  That’s kind of simple, “How’d you get your official public IPv4 addresses that you have today?”  Usually the answer is that you got them from your Internet Service Provider (ISP) such as ATT, Sprint, Comcast, or the like when you had your Internet connection line pulled into your building.  That same concept still applies as the big Internet knows generally where to find you by knowing what region you are in, and what ISP you are connected to by the general range of addressses you are using.

Of course some of you were the lucky ones that actually got a block of IP Addresses early on when IP addresses were being given away just by writing and asking for a block of addresses.  For those of you spoiled by owning your own IPv4 block, you're now at the mercy of your ISP to “loan you” a block of their addresses, you no longer own IP addresses (for IPv6) even if you owned IPv4 addresses before.

However, you CAN get ARIN to issue you your own block of IPv6 addresses if you are an ISP or will be acting as an ISP, see <https://www.arin.net/participate/meetings/reports/ARIN_XV/PDF/sun/IPv6_Huberman.pdf> for a document on how to request IP Addresses from ARIN directly.

**What the equivalent of Private (internal) network addressing is in IPv6**

So the next question that is asked is “how about Private (internal) network addresses, do they exist in IPv6?”, and the answer is Yes.  So if you are just fiddling around with IPv6 in your lab, or you want to do the equivalent of network address translation where you have private addresses for your internal servers and systems, then you can use IPv6 private addressing, or what is called Unique Local Addresses (ULA). In the IPv4 world, private addresses include 10.0.0.0-10.255.255.255, and 172.16.0.0-172.31.255.255, and 192.168.0.0-192.168.255.255.  In the Ipv6 world, the ULA space is fc00::/7, or basically anything that starts with FD in the IPv6 address, so fdxx:xxxx:xxxx…

Bottom of Form

Do note though, if you use Unique Local Addressing in IPv6, these addresses cannot be routed on the Internet.  These devices will always have to remain behind a router (good and bad). Good is that you control these devices like you do IPv4 devices on the “inside” of your network.  So some may say using a ULA is more secure because the device cannot be accessed externally.  However, if everything is on the inside of a firewall, no one can access the device anyway. And because there are so many IPv6 addresses, it’s not like someone will “guess” the address of the destination devices either.

Another argument against Unique Local Addressing is the whole concept of IPv6 is to be able to have IPv6 devices globally routable so that in the future, you want to have your IPv6 systems talk to other IPv6 systems directly without having to translate addresses through a router (from private to public addressing), having publicly accessible IPv6 addresses on internal devices is planning for the future of what will come in IPv6 communications.

This is a tough one.  We got convinced by ISPs to setup Network Address Translation (NAT) and hide everything behind a firewall with non-routable private addresses and we think we have security.  But if we simply use routable IPv6 addresses and create secured subnets protected by routers and firewalls, we’re effectively getting the same security without having to have the overhead of address translation.  I highly encourage organizations to consider implementing publicly addressable IPv6 addresses for all devices.

But if you are fiddling with IPv6 in your lab, rather than using a legally assignable public address (illegally), you might as well use the ULAs and do private internal addressing. When you are fiddling with ULAs, you can pick anything that starts with fd and pick anything you want after that, or if you want a truly randomly generated ULA, there’s a Website that will generate a unique group of ULAs for you <http://www.simpledns.com/private-ipv6.aspx>.  This site will grab 1 of the 72-trillion possible Network/Subnet addresses for you to work with in your lab, that’ll then give you your open realm to use any of the 18-quintillion (18 with 18 zeros) devices connected to your network/subnet.  Again, while ULAs are not routable, if you were to put a ULA addressed device on the Internet, it would likely be unique.

**Understand IPv6 addressing**

Okay, so I’ve gone through the concepts of IP Addressing in IPv6, which basically just said that the same concepts we’ve used and have gotten familiar with in IPv4 still is the same in IPv6, but with slightly different notation.  Now to take a step back and actually provide you details on how IPv6 addressing is done.

In IPv4, as a 32-bit address, we separated the 32-bits into 4 octets separated by periods (or dots), so it looks like 10.12.2.200.  We’d give the address a Subnet mask like 255.255.0.0 which means the network is 10.12 and the device address is 2.200.

In IPv6, as a 128-bit address, rather than breaking into dot separated octets that would end up being 16 numbers (separated by periods) long, IPv6 uses hexadecimals in a double-octet format separated by a colon, so effectly written out as 8 sets of “numbers” (since this is hexadecimal, it is 0 thru 9, and a thru f) so something like   fd30:0000:0000:0001:ff4e:003e:0009:000e

IPv6 addressing allows you to drop preceding zeros in the format, so the above could be simplied as:  fd30:0000:0000:1:ff4e:3e:9:e

And when you have a double-octet group that is nothing but zeros (0000) you can replace the entire grouping with a ::, so this further simplies the above to look like fd30::1:ff4e:3e:9:e  (note:  You can only have 1 set of :: in an IPv6 address, so if you have 2 groupings of zeros, you would put the :: on one set but not on the second set to truncate.

**How to subnet IPv6**

So I told you I’d explain how subnetting works in IPv6.  For the above address, it’s not just one massively long IP address.  It’s actually broken down into 3 parts, the Network Address, the Subnet Address, and the Device Address.

The Network Address is the first 48-bits of the address, or since they are grouped in 16-bit groupings, effectively the first 3 groups of numbers designate the network.  For the above example, the Network Address is fd30:0000:0000.  For those getting their IPv6 addresses from an ISP, the first part of this Network Address will be the same for all of the customer’s of the ISP, which will designate the region and ISP.  If you are doing Unique Local Addressing (ie: IPv6 private addressing), you could effectively just address it as fd00:0000:0000 where fd designates this as a ULA, and that you are working with a single common network.

The Subnet Address is the next 16-bits of the address, or as addresses are grouped in 16-bit groupings, the next group in the IPv6 string.  For the above example, that would be 0001.  Instead of a Subnet Mask, in IPv6, you just note the Network Address and the Subnet Address, and that’ll give this address a specific designation of the Network that this device is on, and the Subnet that this device is one.  This is where I noted IPv6 is more efficient than IPv4 as each packet has everything a router needs to route the information along, instead of having to add or append routing information, or look to a completely separate subnet mask parameter to work backwards into the address.

The last 64-bits (or 4 groupings) is the unique device address, in this case, the device is specifically ff4e:003e:0009:000e.

With 16-bit allocated to subnets, and 64-bits allocated to devices on a subnet, that means a single Network Address can have 65,535 subnets each with over 18-quintillion (18 with 18 zeros after it) devices.  And with 48-bits allocated to the Network Address header of the IPv6 address, that’s 281-trillion networks (with 65,535 subnets, with 18-quintillion devices).

**How the concept of gateways and routing works in IPv6**

So for a Gateway address in IPv6, it works exactly like IPv4, you’ll have some IPv6 address that’ll be the route out of your Subnet.  Just like in IPv4, that Gateway address needs to be an IP address ON the subnet you are on so that your traffic hits that Gateway address, and presumably that Gateway address will then be configured to route your communications to a device outside of your subnet.

Routing works the same too, there will be IPv6 routes that will be dynamically configured, or you will statically configure routes between subnets

Hopefully now the super long IPv6 address makes a little sense.  From you ISP, you will likely be given the first 4 groupings of numbers (the Network and Subnet) and you will have the last 4 groupings to address as you please.  You will define your Gateway address that’ll take you out of your subnet to other subnets.  In the case of a simple router between your subnet and the public Internet, that gateway address will route from your subnet out through your router to your ISP and on to the public Internet.