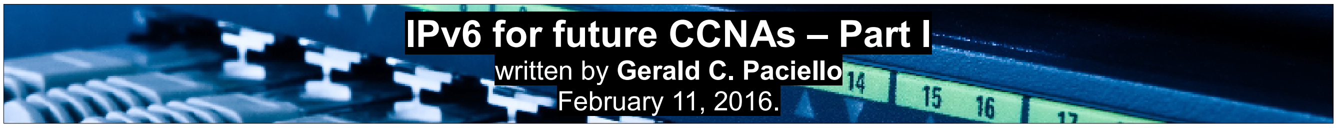

IPv6 for future CCNAs – Part I

written by **Gerald C. Paciello**
February 11, 2016.

The History.

So we are running out (have ran out?) of **IPv4 addresses**. They have been with us since the very beginning, they were there during the 1980`s when everything was calm, and they stuck by us during the chaotic growth of the **Internet** during the 1990`s. There were some rumors, during the 2000`s, that they were going to run out soon, but something called **Classless Inter-Domain Routing (CIDR)** and **Network Address Translation/Port Address Translation (NAT/PAT)** saved our butts... for a while.

These 4 little sets of numbers, called **Octets,** separated by dots were very mysterious, we used to look at them from the corner of our eyes asking our selves; What are they for?... What do they do?... How do they do it?
We were intimidated just by looking at them, and they were just 4 little sets of decimal numbers, numbers that we recognize, that we use every day in our daily lives.

And then, we found out that they were hiding something, they were just **expressed** in a decimal format (called **dotted-decimal)**, when in reality they were written in something called **Binary,** and that they were not 4 little sets of decimal numbers but 4 **octets** with **8bits** each for a total of **32** binary **1**s and **0**s!... and that´s not all, there were **2^32** of them, which is **4,294,967,296.** Oh my God! What is going on?...

But time went by and we, the "**network guys**", in the long run became accustom to these little guys. We learned that they uniquely identify a device on the network. That they have a friend, called the **Sub-net Mask** that indicates to what network they belong to, we've even learned to break them in small chunks according to the size of our network, using **sub-netting**, so we wouldn't waste them... we were taking care of them.

Regardless, no matter what we do to preserve them, we were going to run out soon enough. Something had to be done... Introducing, **Internet Protocol Version 6**... and get this, it is **NOT** a string of **32** but **128** binary **1**s and **0**s! It is **NOT** 4 little fields separated by dots, it´s 8 **16bits** fields called **Hextets,** separated by colons (**:**)! It is **NOT** expressed in decimal format but in **Hexadecimal** format!.. and there are not **2^32** but **2^128** of them, which is... are you ready?... **340,282,366,920,938,463,463,374,607,431,464,133,816** (for those of you who are just starting out and don´t know, the number is **340 undecillion...** plus change)

I mean, I understand that the world cannot continue in its current form without the **Internet,** but come on!! What are they trying to do to us?

Advantages Over IPv4.

As you know, the migration over to **IPv6** has already started, but you haven't noticed it maybe, because **IPv4** is still around and it will still be around for a long time. There are many reasons why IPv4 is still around; **NAT/PAT**, **CIDR** and also, it works fine and we know how it works.
But, we also know that there are no more **IPv4** addresses to give out and this is the main reason why we have to migrate to **IPv6.**
But what about the people that already have IPv4 addresses assigned, I mean they have them for good right, so what makes them want to migrate to **IPv6**?
The answer is because not only there is an almost infinite amount of addresses available in **IPv6 (**and we will **NEVER** ran into the same shortage problem again... **right?**), but also because **IPv6** has some pretty good **advantages** over **IPv4**.

Let´s see a few:

- **Better End-to-End Connectivity:**
  - One of the features that allowed **IPv4** to exist beyond what was expected, is **NAT**. However, **NAT** is not a very good **point-to-point** communicator. **Host A** thought it was talking directly with **Host B** when in fact, it wasn't.
    Well, **IPv6**, because of its vast address space size, no longer needs the use of **NAT**. It allows for **direct end-to-end communication from Host A to Host B**... no middle man.

- **Better Auto-configuration.**
  - For a host to get the info needed to join a network, **IPv4** uses **DHCP** (**Dynamic Host Config Protocol**) which is a ***stateful*** method, meaning the host receives all the info from a server.
    IPv6 has both a **stateful** (**DHCPv6**) and ***stateless*** methods where hosts are able to **auto-configure themselves** with the info needed to join the network (no server).

- **More Efficient Header:**
  - There is a significant improvement on processing time because many rarely used fields from the **IPv4** header have been either removed, or moved to an optional header called "***extension Header***". This extension header is only implemented by intermediate routers if a packet needs special handling.
    One of the fields that have been removed is the **Checksum** field. Routers no longer have to compute a checksum every time they receive a packet. With IPv6, checksum and error control is handled by upper-layer protocols.

- **Better Security.**
  - IPv6 has support for IPSec which provides for Data Confidentiality, Integrity and Authentication at layer 3. With IPv4, end devices provided this level of security.

- **Better QoS Support.**
  - This is accomplished through the use of a field in the header called **Flow Label**. Routers are able to use this field to mark specific flows of packets such as packets that require **QoS** treatment.

- **Built-in Mobility Support:**
  - IPv6 hosts have the ability to move around the network and maintain its IP address.

- **Transition Tools.**
  - Tunneling
  - NAT64
  - 6to4
  - SIIT
  - DNS64
  - etc.

IPv6 address format.

An **IPv6** address consists of **32 hexadecimal** numbers, separated by **colons** (**:**) into **8 hextets** of **4** hex numbers each. Each hex number represents **4 bits,** that is **16bits** per **hextet** for a total of **128bits** (**4**bits x **4** hex digits per hextet x **8** hextets= **128bits**).

Here is an example:

2001:0AC8:1234:0000:0000:0000:0000:0678

Let´s break it down into binary. We are not going to do the whole number, but lets do the 1st and 2nd hextets at least:

Hextets	1st				2nd			
Hex	2	0	0	1	0	A	C	8
Place Values	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1	8 4 2 1
Binary	0 0 1 0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 0 0	1 0 1 0	1 1 0 0	1 0 0 0

Remember, to figure out binary, you need to use the **place values,** and they are; **8 4 2 1** for a 4 bit value like in

this case (each hex digit).
So, for example, hex number **A** on the second hextet (which is 10 in decimal), there is a **0** on place value **1**, a **1** on place value **2**, a **0** on place value **4** and another **1** on place value **8**. Now just add the values that have **1**s, **2+8=10.**

IPv6 Short Notation.

As you can see, an **IPv6** address is very long, right? Let´s imagine this scenario: You arrive at work and there´s an email from your boss, asking you very nice and politely... sort of, that all **100 PCs** for the upcoming event are in *Show Room C*, and that you need to configure these **PCs,** for some odd reason, with **IPv6 addresses... manually**.
Well... your next step should be texting (texting, is that an obsolete word now?) your wife to let her know that you are not going to get home on time this evening, am I correct?

Fortunately, some genius people already thought about (or went through!) this scenario, and they came up with a way to be able to write an **IPv6** address in a much shorter way. Let´s see it.

First, you need to understand the rules, they are very simple:

1. Leading zeroes on each hextet can be omitted. Leading zeroes **only**. So:
   - `2001:0AC8:1234:0000:0000:0000:0000:0678` can be written as:
   - **`2001:AC8:1234:0:0:0:0:678`**
2. Contiguous hextets of zeroes, can be represented with the use of **double colons** (**: :**). This can ***only be implemented one time per address***. So:
   - `2001:0AC8:1234:0000:0000:0000:0000:0678` can be written as:
   - **`2001:0AC8:1234::0678`**
3. And finaly, we can **combine rules 1 and 2**. So:
   - `2001:0AC8:1234:0000:0000:0000:0000:0678` can be written as:
   - **`2001:AC8:1234::678`**

Also, we can still use "**Slash Notation**" or "**CIDR Notation**" as we did with **IPv4**. For example, if the first **64** most signifficant bits indicate the network bits or **Network ID**, we can notate it with a slah (**/**) and the number of bits i the network ID, like this ***/64***.

So, we can write the whole address as follows:

`2001:AC8:1234::678/64`

On this particular example, we ended up with a shorter **IPV6** address, only 14 hex digits as opossed to 32 in a full address. However, depending on the original, full address, we can end up with an address that is only 10 characters long, and that is very good if you need to enter these addresses manually. Lets see:

Original: **2001:0000:0000:0000:0000:0000:0000:0001/64**
Let`s remove the leading zeroes: **2001:0:0:0:0:0:0:1/64**
Or, we can us the double colon: **2001::1/64**

IPv6 Address Types.

In IPv4 we have **3** types of addresses; **Unicast**, **Multicast** and **Broadcast.** For IPv6, even though there is no more Broadcast address, there are several types of addresses and they are assified within 3 main types.
They are:

- **Unicast**
  - **One-to-one** communication. **Unique** address assigned to an interface, a packet sent to a Unicast address will be received by **one single interface**. There are several types of Unicast addresses:
    - **Global Unicast.**

- **Link Local.**
- **Unique Local.** (in place of Site-Local which was deprecated in 2004)
- **Unspecified.**
- **Loopback.**

- **Multicast**
  - **One-to-many** communication. A **Multicast** address identifies a **group of interfaces**. A packets sent to a **Multicast** address are received by a group of interfaces that may be in different hosts.

- **Anycast**
  - **Special one-to-one** communication. An **Anycast** address represents a **group of interfaces**, but the packet sent to this address will be deliver **only** to the interface which is **closest**, in terms of the **routing protocol cost value**.
    Also, since **Anycast** addresses are allocated from the **Unicast address space**, they are syntactically indistinguishable from each other. So, an Anycast address is a Unicast address that was assigned to more than one interface.

END OF PART I

References:

- Members of **The Cisco Learning Network** at https://learningnetwork.cisco.com/welcome
- IPv6 Fundamentals: A Straightforward Approach by **Rick Graziani**
- https://docs.oracle.com/cd/E19683-01/817-0573/chapter1-26/index.html
- https://technet.microsoft.com/en-us/library/bb726995.aspx#EDAA
- http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml
- http://ipv6.com