| Winsock |< Linux/Unix OS Security Features |Wi-fi Security Features >| Main | Site Index | Download |

Internet Protocol version 6 (IPv6)

What do we have in this Module?

1. INTRODUCTION

The main point the IPv6 (Internet Protocol version 6) is introduced and implemented to overcome the expected shortage of IP addressing using IPv4 (Internet Protocol version 4). It is designed to support continued Internet growth in number of users and functionality. Currently we are in transition period, using the mixed of IPv4 and IPv6. With the explosion of the new devices that can be attached to the Internet such as consumer electronics gadgets, the demand for new Internet Protocol addressing is unavoidable.

Other than more addressable IPs, there are many new features and improvements have been incorporated in IPv6. It is a new layer 3 protocol of the 7 layer OSI [1]; hence the need of new mechanism or properties to coexist with the current TCP/IP stack. However new version of the related protocols in the TCP/IP stack such as ICMP6 already developed as well.

IPv4 was designed long time ago [2] around January 1980 and since its introduction, there have been many requests for more addresses and enhanced capabilities. The major changes in IPv6 are the redesign of the header, including the increase of address size from 32 bits to 128 bits. Because layer 3 is responsible for end-to-end packet transport using packet routing based on addresses, it must include the new IPv6 addresses of the source and destination similar to IPv4.

2. THE MOTIVATIONS

Tremendous growth may be the basic issue which caused the need for a new generation IP, the IPv6. Previously, only the computer market has been the driver of the Internet growth. This market has been growing at an exponential rate.

The computers which are used at the endpoints of internet communications range from PCs to Supercomputers. Most are attached to Local Area Networks and the huge majority is not mobile. While the computer market will continue to grow at significant rates due to expansion into other non-traditional areas it is doubtful it will continue to grow at an exponential rate.

New mass markets that fall into consumer market already developed. This market demand a new set of requirements which were not as evident in the early stages of IPv4 deployment. For example the nomadic personal computing devices such as PDA, GPS and hand phones already can be seen everywhere as their prices drop and their functionalities enhanced. They will support a variety of network attachment types. For instance, when disconnected from wired network, they will use RF wireless networks and when used in networked facilities they will use infrared attachment.

In addition to the obvious requirement of an internet protocol which can support large scale routing and addressing, they will require an internet protocol which imposes a low overhead, supports auto configuration and mobility as basic properties. From the security aspect, the nature of roaming computing devices requires an internet protocol to have built in authentication and confidentiality.

Another market is networked entertainment and edutainment such as hundred world television channels via satellite (satellite TV), Internet TV and video on demand. Later on we can see that every television set will become an Internet host with fixed IP. Furthermore, as the world of digital high definition television approaches, we will see that the differences between a computer and a television will diminish.

This market will require an Internet protocol which supports large scale of routing and addressing, and auto configuration. They also require a protocol suite which imposes the minimum overhead for efficiency and ease of use. Similar to the device control that used to control of everyday devices such as lighting equipment, CCTV, air conditioning, fridges, and other types of equipment which are currently controlled via analog. The size of this consumer market is enormous and requires solutions which are simple, robust, easy to use, and very low cost.

The using of the smartcard as a human identification with personal and multiple services information is another potential candidate for IP address assignments. In this case, every human in the world will have personal IP address in the future, embedded in the smartcard or similar.

Global internet routing based on the on 32-bit addresses of IPv4 is becoming increasingly strained. IPv4 addresses do not provide enough flexibility to construct efficient hierarchies which can be combined. The deployment of Classless Inter-Domain Routing (CIDR) [3] is extending the life time of IPv4 routing by a number of years however the effort to manage the routing will continue to increase. Even if the IPv4 routing can be scaled to support a full IPv4 Internet, the Internet will sooner or later run out of network numbers.

3. FEATURES

As a new version of the Internet Protocol, IPv6 was designed as a successor to IP version 4 [2]. IPv6 is assigned IP version number 6 and is formally called IPv6 [4]. Functions which work in IPv4 were kept in IPv6 and which didn't work was removed.

The following are the main list of IPv6 changes compared to IPv4.

3.1 Expanded Routing and Addressing Capabilities

IPv6 increases the IP address size from 32 bits to 128 bits, to support a much greater number of addressable nodes and more levels of addressing hierarchy, and simpler auto-configuration of addresses. The scalability of multicast routing is improved by adding a scope field to multicast addresses. With no hidden networks and hosts, all hosts can be reachable and be servers, enable the global reachability. The use of 64 bits for link-layer addresses encapsulation with warranty of uniqueness.

A new type of address called an anycast address is defined, to identify sets of nodes where a packet sent to an anycast address is delivered to one of the nodes. The use of anycast addresses in the IPv6 source route allows nodes to control the path which their traffic flows. Mandatory features include security such as IP Security (IPSec) for mobility which optimized in IPv6 than mobile IPv4. There are no broadcasts, providing an efficient use of the network and less interrupts on NICs. IPv6 provides a much larger pool of multicast addresses with multiple scoping options.

3.2 Header Format Simplification

Some IPv4 header fields have been dropped or made optional, to reduce the common-case processing cost of packet handling and to keep the bandwidth cost of the IPv6 header as low as possible despite the increased size of the addresses. Even though the IPv6 addresses are four time longer than the IPv4 addresses, the IPv6 header is only twice the size of the IPv4 header.

3.3 Options Improvement

IPv6 options are placed in separate headers that are located between the IPv6 header and the transport layer header. Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future. Less number of fields such as no checksum enables routing efficiency, performance, forwarding rate scalability and with the extensibility of header will provide better handling of options.

3.4 Quality-of-Service (QoS)

A new flow label capability is added to enable the labeling of packets belonging to particular traffic flows for which the sender requests special handling, such as non-default quality of service or real- time service.

3.5 Authentication and Privacy

IPv6 includes the definition of extensions which provide support for authentication, data integrity, and confidentiality. This is included as a basic element of IPv6 and will be included in all implementations.

3.6 Auto Configuration

Using IPv4 addresses, clients use the Dynamic Host Configuration Protocol (DHCP) server to request an address each time they log into a network. This address assignment process is called stateful auto-configuration. IPv6 supports a revised DHCPv6 protocol that supports stateful auto-configuration, and supports stateless auto-configuration of nodes.
Stateless auto-configuration does not require a DHCP server to obtain addresses. Stateless auto-configuration uses router advertisements to create a unique address. This creates a plug-and-play environment, simplifying address management and administration such as for multi-homing. IPv6 also allows automatic address configuration and reconfiguration. This capability allows administrators to re-number network addresses without accessing all clients.

4. IPv6 HEADERS

Let take a look at the IPv6 packet level. Based on the IETF's RFC 2460 [4] the following is the IPv6 header format. Without extension, IPv6 header is 40 bytes.

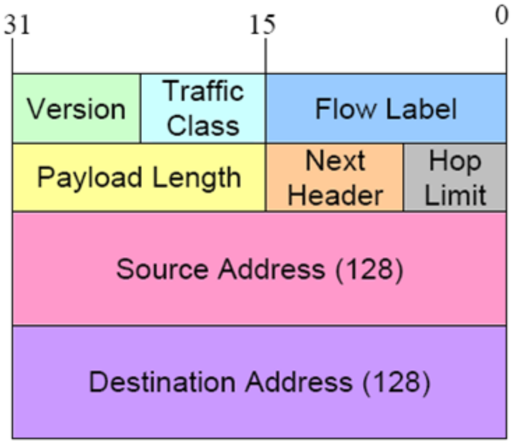

Figure 1. IPv6 Header Format

Table 1. IPv6 Headers' Elements

Element	Description
Version	A 4-bit Internet Protocol version number = 6.
Traffic Class	An 8-bit traffic class field. Identifies different classes or priorities (diffserv).
Flow Label	A 20-bit flow label. Used by a source node to label sequences of packets
Payload Length	A 16-bit unsigned integer. Length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, i extension headers [section 4] present are considered part of the payload, i.e., included in the length count.)
Next Header	An 8-bit selector. Identifies the type of header immediately following the IPv6 header. Uses the same values field. Used to identify the encapsulated protocol - TCP, UDP, ESP, AH (confidentiality and authentication in IF extensions.
Hop Limit	An 8-bit unsigned integer. Decremented by 1 by each node that forwards the packet. The packet is discarde decremented to zero. It is TTL in IPv4.
Source Address	A 128-bit address of the originator of the packet.
Destination Address	A 128-bit address of the intended recipient of the packet (possibly not the ultimate recipient, if a Routing hea

4.1 Extension Headers

In IPv6, optional internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper-layer header in a packet. This new way of implementing the options is added after the basic IPv6 header by daisy chained method. There are a small number of such extension headers, each identified by a distinct Next Header value. An IPv6 packet may carry zero, one, or more extension headers, each identified by the Next Header field of the preceding header. The following Figure is an illustration of the extension headers.

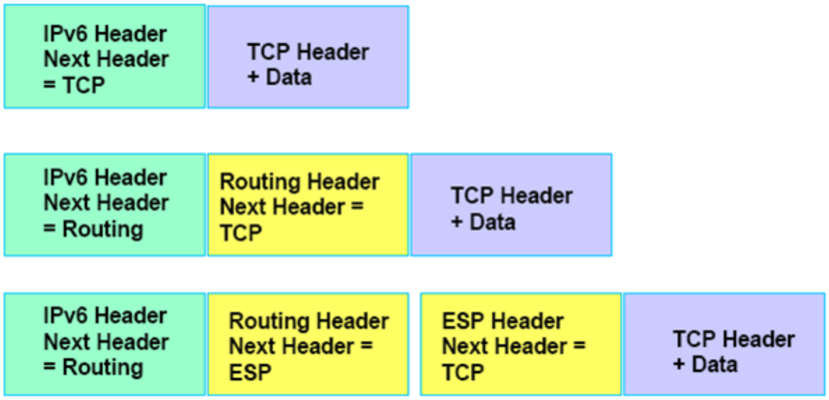
IPv6 Header Next Header = TCP
TCP Header + Data
IPv6 Header Next Header = Routing
Routing Header Next Header = TCP
TCP Header + Data
IPv6 Header Next Header = Routing
Routing Header Next Header = ESP
ESP Header Next Header = TCP
TCP Header + Data

Figure 2. IPv6 Extension Headers Example

Each extension header is an integer multiple of 8 octets long (1 octet = 8 bits), in order to retain 8-octet alignment for subsequent headers. Multi-octet fields within each extension header are aligned on their natural boundaries, that is, fields of width n octets are placed at an integer multiple of n octets from the start of the header, for n = 1, 2, 4, or 8. A full implementation of IPv6 includes implementation of the following extension headers.

1. IPv6 header.
2. Hop-by-Hop Options header.
3. Destination Options header (when the routing header is used).
4. Routing header.
5. Fragment header.
6. Authentication header.
7. Encapsulating Security Payload header.
8. Destination Options header.
9. Upper-layer header.

Source node should follow this order, but destination nodes should be prepared to receive them in any order. When more than one extension header is used in the same packet, it is recommended that those headers appear in the following order after the basic IPv6 header:

1. Hop-by-Hop Options.
2. Routing (Type 0).
3. Fragment.
4. Destination Options.
5. Authentication.
6. Encapsulating Security Payload.

The first four are specified in [4]; the last two are specified in [5] and [6], respectively. The following is a summary of the extension headers.

Table 2. IPv6 Extension Headers

Extension header	Description
Hop-by-hop options (0)	Information that must be examined by every node along the path. It is used by Router Alert and Jumbogram.
Routing (43)	It is similar to IPv4's Loose Source and Record Route option. Used by mobileIPv6.
Fragment (44)	Used by source node.
Destination options (60)	Used to carry optional information that need to be examined only by a packet's destination node(s). Used by MobileIPv6.
Authentication (AH)	The IP Authentication Header (AH) is used to provide connectionless integrity and data origin authentication for IP datagrams, and to provide protection against replays.
Encapsulating Security Payload (ESP)	To provide a mix of security services in IPv4 and IPv6. ESP may be applied alone, in combination with the IP Authentication Header (AH), or in a nested fashion, e.g., through the use of tunnel mode.

Comparison of IPv4 and IPv6 headers shows a longer header, but less number of fields. The header processing is simpler, the options are handled by extension headers and routing header for source routing changes the destination address in the IP header.

5. IPv6 ADDRESSING

Similar to IPv4, IPv6 addresses can be split into network and host parts using subnet masks. IPv4 has shown that sometimes it

would be nice, if more than one IP address can be assigned to an interface, each for a different purpose such as for aliases and multi-cast. To remain extensible in the future, IPv6 is going further and allows more than one IPv6 address to be assigned to an interface. There is currently no limit defined by an RFC, only in the implementation of the IPv6 stack.
Using this large number of bits for addresses, IPv6 defines address types based on some leading bits, which are hopefully never going to be broken in the future unlike classless type IPv4 and the history of class type A, B, and C addresses. Also the number of bits are separated into a network part (upper 64 bits) and a host part (lower 64 bits), to facilitate the auto-configuration.
The design of the address types left a lot of scope for future definitions as currently there are many more unknown or undecided requirements. [10] defines the current addressing scheme. Now lets take a look at the different types of prefixes and therefore address types available in IPv6. In the meantime we will try to compare it with similar things in IPv4. IPv6 addresses are 128 bits long. This number of bits generates a very big decimal numbers with up to 39 digits as shown below.

$2^{128} - 1 = 340282366920938463463374607431768211455$

A better notation for such big numbers is using hexadecimal representation. In hexadecimal, 4 bits (also known as nibble) are represented by a digit or character from 0-9 and A-F (10-15). This format reduces the length of the IPv6 address to 32 characters.

$2^{128} - 1 = $ 0xffffffffffffffffffffffffffffffff

To make it more convenient, a hexadecimal format with a colon as separator after each block of 16 bits is chosen for IPv6 representation. This look likes similar to IPv4 that uses dot as separator. In addition, the leading "0x", a signifier for hexadecimal values used in programming languages is removed. So we have (all 1s):

$2^{128} -1 = $ ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

A usable IP address example is shown below:

```
2351:0ca8:0300:a171:0617:b4f5:fac3:782a
```

For simplifications, leading zeros of each 16 bit block can be omitted. Then for the following address:

```
2351:0ca8:0300:a171:0617:b4f5:fac3:782a
```

Can be simplified to

```
2351:ca8:300:a171:617:b4f5:fac3:782a
```

One sequence of 16 bit blocks containing only zeroes can be replaced with "::". But not more than one at a time, otherwise it is no longer a unique representation. For example:

```
2351:0ca8:100:f101:0:0:0:1
```

Becomes

```
2351:ca8:100:f101::1
```

The biggest number of digit reduction is seen by the IPv6 localhost address:

`0000:0000:0000:0000:0000:0000:0000:0001` becomes `::1`

You can use ipv6cal, an IPv6 address format calculator and converter program at [7] to calculate the IPv6 addresses.

5.1 Addresses Without a Special Prefix

We have Unicast, Multicast and Anycast address types in IPV6 (no broadcast as in IPv4). In the Unicast, we have several categories as listed below and will be explained in the following sections:

1. Unspecified
2. Loopback
3. Scoped addresses:
    a. Link-local
    b. Site-local
4. Aggregatable Global.

5.1.1 Unspecified Address

The Unspecified used as a placeholder when no address available. For example for initial DHCP request and Duplicate Address Detection (DAD). It is like 0.0.0.0 in IPv4. So in IPv6 we have 0:0:0:0:0:0:0:0 or :: These addresses are mostly used in socket binding (to any IPv6 address) or routing tables. Take note that the unspecified address cannot be used as destination address. For IPv6 it is:

```
0000:0000:0000:0000:0000:0000:0000:0000
```

Or can be compressed to: `::`

5.1.2 The Localhost Address

Loopback used to identifies self and it is Localhost IP address. This is a special address for the loopback interface, similar to IPv4 with its 127.0.0.1. With IPv6, the localhost address is:

```
0000:0000:0000:0000:0000:0000:0000:0001
```

Or can be compressed to `::1`

As in IPv4, packets with this address as source or destination should never leave the sending host and cannot be routed.

5.1.3 IPv6 Address With Embedded IPv4 Address

There are two addresses which contain an IPv4 address.

1. IPv4-mapped IPv6 address.

   IPv4-only IPv6-compatible addresses are sometimes used for sockets created by an IPv6-enabled process (daemon or service), but only binding to an IPv4 address. These addresses are defined with a special prefix of length 96 (`a.b.c.d` is the IPv4 address):

   ```
   0:0:0:0:0:ffff:a.b.c.d/96
   ```

   Or in compressed format: `::ffff:a.b.c.d/96`

   For example, the IPv4 address `1.2.3.4` looks like this:

   ```
   ::ffff:1.2.3.4
   ```

2. IPv4-compatible IPv6 address

   This address used for automatic tunneling [8], which is being replaced by 6to4 tunneling [9]. The example is:

   ```
   0:0:0:0:0:0:a.b.c.d/96
   ```

   Or can be compressed to

   ```
   ::a.b.c.d/96
   ```

5.2 Network Part or Prefix

5.2.1 Link Local Address Type

Link-locals are special addresses (a scoped address – the scope is local link such as Virtual Local Area Network, VLAN and subnets) is new in IPv6 which will only be valid on a link of an interface. It is automatically configured on each interface by using the interface identifier based on Media Access Control (MAC) address. Using this address as destination the packet would never pass through a router. It gives every node an IPv6 address to start communications. It is used for link communications such as:

- Anyone else here on this link, that is in the same subnet?
- Anyone here with a special address such as finding routers?

They begin with (where “x” is any hex character, normally “0”) as shown below. An address with this prefix is found on each IPv6-enabled interface after stateless auto-configuration which is normally always the case.

```
fe8x:
```

```
fe9x:
feax:
febx:
```

The format will look something like:

```
FE80:0:0:0:<interface identifier>
```

5.2.2 Site Local Address Type

Site locals are addresses similar to the RFC 1918 [11], scoped address in IPv4 today (the scope is site or a network of link), with the added advantage that everyone who use this address type has the capability to use the given 16 bits for a maximum number of 65536 subnets, enabling an addressing plan for a full site. It is comparable to the 10.0.0.0/8 in IPv4 today. Another advantage, it is possible to assign more than one address to an interface with IPv6, you can also assign such a site local address in addition to a global one. The addresses begin with:

```
fecx:
fedx:
feex:
fefx:
```

Where “x” is any hex character, normally “0”. This address type is now deprecated [12]. These addresses can only be used between nodes of the same site and cannot be routed outside the site such as the Internet. It is very similar to IPv4 private addresses and by default it is not configured. The format is:

```
FEC0:0:0:<subnet id>:<interface id>
```

5.2.3 Unique Local IPv6 Unicast Addresses

Because originally defined site local addresses are not unique, this can lead to major problems, if two former independent networks would be connected later, that is the subnets are overlapped. This and other issues lead to a new address type named [13]. These addresses begin with:

```
fdxx:
fcxx:
```

A part of the prefix (40 bits) is generated using a pseudo-random algorithm and it is oddly, that two generated ones are equal. An example for a prefix generated using a web-based tool [14] is:

```
fd0f:8b72:ac90::/48
```

5.2.4 Global Address Type (Aggregatable Global Unicast)

It is reserved for generic use and globally reachable. It is allocated by Internet Assigned Numbers Authority (IANA) to Regional Registries [17], and then tier-1 Providers normally called **Top-level Aggregator** (TLA). Next allocated to Intermediate Providers normally called **Next-level Aggregator** (NLA), then to sites and finally to subnets. One global address type defined (the first design, called "provider based," was obsolete years ago [15], however you may find some still remains in older Linux kernel sources for example). It begins with (x are hex characters)

```
2xxx:
3xxx:
```

However, there are some further subtypes defined, as explained in the following section.

5.2.4.1 6bone Test Addresses

These were the first global addresses which were defined and in use. They all start with:

```
3ffe:
```

For example:

```
3ffe:ffff:100:f102::1
```

A special 6bone test address (obsolete) which will never be globally unique begins with:

```
3ffe:ffff:
```

and is mostly shown in older examples, because if real addresses are shown, it's possible for someone to do a copy & paste to their configuration files. Thus inadvertently causing duplicates on a globally unique address. This would cause serious problems for the original host such as getting answer packets for request that were never sent. Because IPv6 is now in production, this prefix is no longer be delegated and removed from routing after 6.6.2006 [16].

5.2.4.2 6to4 Addresses

These addresses, designed for a special tunneling mechanism [9] and [8], encode a given IPv4 address and a possible subnet and begin with:

```
2002:
```

For example, representing `192.168.1.1/5`:

```
2002:c0a8:0101:5::1
```

5.2.4.3 Assigned by Providers

These addresses are delegated to Internet service providers (ISP) and begin currently with:

```
2001:
```

Prefixes to major (backbone owner) Internet Service Providers (ISPs) are delegated by local registries [17] and currently they got a prefix with length 32 assigned. Any ISP customer can get a prefix with length 48.

5.2.4.4 Reserved Addresses

Currently, two address ranges are reserved for examples and documentation [18]. These addresses are:

```
3fff:ffff::/32
2001:0DB8::/32
```

These address ranges should be filtered based on source addresses and should not be routed on border routers to the internet, if possible.

5.3 Multicast Addresses

Multicast is one-to-many type addresses and is used for related multicast services. There is no broadcast in IPv6, multicast is used instead. They always start with (xy is the scope value):

```
ffxy:
```

They are split into scopes and types explained in the following sections.

5.3.1 Multicast Scopes

Multicast scope is a parameter to specify the maximum distance a multicast packet can travel from the sending entity. Currently, the following scopes are defined:

- `ffx1`: node-local, packets never leave the node.
- `ffx2`: link-local, packets are never forwarded by routers, so they never leave the specified link.
- `ffx5`: site-local, packets never leave the site.

- `ffx8`: organization-local, packets never leave the organization (not so easy to implement, must be covered by routing protocol).
- `ffxe`: global scope.
- others are reserved

So, it is node, link, site, organization and global scopes. The format is `FF<flags><scope>::<multicast group>`

5.3.2 Multicast Types

There are many types already reserved [10]. Some examples are:

- All Nodes Address: ID = 1h, addresses all hosts on the local node (ff01:0:0:0:0:0:0:1) or the connected link (ff02:0:0:0:0:0:0:1).
- All Routers Address: ID = 2h, addresses all routers on the local node (ff01:0:0:0:0:0:0:2), on the connected link (ff02:0:0:0:0:0:0:2), or on the local site (ff05:0:0:0:0:0:0:2)

5.3.3 Solicited Node Link-local Multicast Address

Special multicast address used as destination address in neighborhood discovery, because unlike in IPv4, Address Resolution Protocol (ARP) no longer exists in IPv6. An example of this address looks like:

```
ff02::1:ff00:1234
```

The used of prefix shows that this is a link-local multicast address. The suffix is generated from the destination address. In this example, a packet should be sent to address “fe80::1234”, but the network stack doesn't know the current layer 2 MAC address. It replaces the upper 104 bits with “ff02:0:0:0:0:1:ff00::/104” leaving the lower 24 bits untouched. This address is now used on-link to find the corresponding node which has to send a reply containing its layer 2 MAC address.

5.4 Anycast Addresses

Anycast is one-to-nearest node and it is good for discovery functionalities. These addresses indistinguishable from unicast addresses because they are allocated from the unicast addresses space and some anycast addresses are reserved for specific uses such as router-subnet, mobile IPv6 home-agent discovery and in DNS discovery.
Anycast addresses are special addresses and are used to cover things like nearest DNS server, nearest Dynamic Host Configuration Protocol (DHCP) server, or similar dynamic groups. Addresses are taken out of the unicast address space (aggregatable global or site-local at the moment). The anycast mechanism from client view will be handled by dynamic routing protocols. Anycast addresses cannot be used as source addresses, they are only used as destination addresses.

5.4.1 Subnet-router Anycast Address

A simple example for an anycast address is the subnet-router anycast address. Assuming that a node has the following global assigned IPv6 address:

```
2001:db8:100:f101:210:a4ff:fee3:9566/64
```

It is node’s address. The subnet-router anycast address will be created blanking the suffix, the least significant 64 bits completely. For example the following is subnet-router anycast address:

```
2001:db8:100:f101::/64
```

5.5 Address Types for Host Part

For auto-configuration and mobility issues, it was decided to use the lower 64 bits as host part of the address in most of the current address types. Therefore each single subnet can hold a large amount of addresses. This host part can be inspected differently:

5.5.1 Stateless (Automatically Computed)

With auto-configuration, the host part of the address is computed by converting the MAC address of an interface (if available), with the EUI-64 method, to a unique IPv6 address. If no MAC address is available for this device such as on virtual devices, something else like the IPv4 address or the MAC address of a physical interface is used instead. For example NIC has following MAC address (48 bit):

```
00:10:A4:E3:95:66
```

This would be expanded according to the [19] design for EUI-48 identifiers to the 64 bit interface identifier:

```
0210:a4ff:fee3:9566
```

With a given prefix, the result is the IPv6 address shown in example above:

```
2001:0db8:0100:f101:0210:a4ff:fee3:9566
```

It is obvious that any IPv6 node should recognize the following addresses as identifying itself:

1. Link-local address for each interface.
2. Assigned unicast/anycast addresses manually or automatically.
3. Loopback address.
4. All-nodes multicast address.
5. Solicited-node multicast address for each of its assigned unicast and anycast address
6. Multicast address of all other groups to which the host belongs

And for routing, any IPv6 router should recognize the following addresses as identifying itself:

1. All the required node addresses.
2. All routers multicast addresses.
3. Specific multicast addresses for routing protocols.
4. Subnet-router anycast addresses for the interfaces configured to act as forwarding interfaces.
5. Other anycast configured addresses.

6. CONCLUSION

This paper describes the IPv6 fundamentals, the formats, the type and scope of the IP addresses. Most of the IP addresses explained and compared to the IPv4 so that we can appreciate the similarities, differences and new features.

From the user perspective, it is obvious that not so much new things we can see other than the new IP address formats that normally transparent to them.

The developments and implementations by vendors and standard bodies provide seamless migration from IPv4 to IPv6. Most of the time, users will not aware the migration processes. For example, when installing the network card, the IPv6 will automatically setup and configured. Another important part of the IPv6 new implementation is the routing and addressing which is being done by the networking devices vendors and the standard bodies.

Hopefully when IPv6 fully implemented, we will have enough IP addresses with new functionalities mainly the security aspects, providing safer mobile communication. It look likes there are many more improvements being implemented based on the new requests of the RFC documents.

7. REFERENCES

[1] OSI model, from Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/OSI_model
[2] RFC 760, DoD standard Internet Protocol, http://www.rfc-zone.org/rfc760.html
[3] RFC 1519, Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy, http://www.ietf.org/rfc/rfc1519.txt
[4] RFC 2460, Internet Protocol, Version 6 (IPv6) Specification, http://www.ietf.org/rfc/rfc2460.txt
[5] RFC 2402, IP Authentication Header, http://www.rfc-editor.org/rfc/rfc2402.txt
[6] RFC 2406, IP Encapsulating Security Payload (ESP), http://www.ietf.org/rfc/rfc2406.txt
[7] The ipv6calc homepage, http://www.deepspace6.net/projects/ipv6calc.html
[8] RFC 2893, Transition Mechanisms for IPv6 Hosts and Routers, http://www.ietf.org/rfc/rfc2893.txt
[9] RFC 3056, Connection of IPv6 Domains via IPv4 Clouds, http://www.ietf.org/rfc/rfc3056.txt
[10] RFC 4291, IP Version 6 Addressing Architecture, http://www.ietf.org/rfc/rfc4291.txt
[11] RFC 1918, Address Allocation for Private Internets, http://www.ietf.org/rfc/rfc1918.txt
[12] RFC 3879, Deprecating Site Local Addresses, http://www.ietf.org/rfc/rfc3879.txt
[13] RFC 4193, Unique Local IPv6 Unicast Addresses, http://www.ietf.org/rfc/rfc4193.txt
[14] Locally Assigned Global ID calculator, http://forschung.goebel-consult.de/ipv6/createLULA
[15] RFC 1884, IP Version 6 Addressing Architecture (obsolete), http://www.ietf.org/rfc/rfc1884.txt
[16] RFC 3701, 6bone (IPv6 Testing Address Allocation) Phaseout, http://www.ietf.org/rfc/rfc3701.txt
[17] Regional/Local Internet Registries information, http://www.ripe.net/info/resource-admin/
[18] RFC 3849, IPv6 Address Prefix Reserved for Documentation, http://www.ietf.org/rfc/rfc3849.txt
[19] IEEE Standard, GUIDELINES FOR 64-BIT GLOBAL IDENTIFIER (EUI-64) REGISTRATION AUTHORITY, http://standards.ieee.org/regauth/oui/tutorials/EUI64.html

| Winsock |< Linux/Unix OS Security Features |Wi-fi Security Features >| Main | Site Index | Download |