Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.829 Fall 2002 | **IPv6 Tutorial Notes** | September 27, 2001

---

1 Problems with IPv4 addressed in IPv6

1. Not Enough addresses. In theory there are 2^{32} total, but allocation isn't very efficient. There are probably 10^8 ($\pm$ a factor of 10 or so) internet hosts today, and growing exponentially.
2. Not enough networks. Even with CIDR addressing, allocating (and then routing) to an ever increasing number of addresses is hard.
3. Simplify Header contents (make router's job easier)
4. Make host configuration easier (No need for DHCP servers in most cases)
5. Address security, mobility, multicast, and other application needs.

2 Header Differences

IPv4:

4 bits: version ==4
4 bits: header length
8 bits: type of service
16 bits: total length
16 bits: identification (for fragments)
1 bit: Don't fragment
1 bit: More Fragments
13 bits: fragment offset
8 bits: Time to Live
8 bits: protocol
16 bits: header checksum
32 bits: source address
32 bits: destination address
0–320 bits: options

Total of 20

IPv6:

4 bits: version == 6
8 bits: traffic class
20 bits: flow label
16 bits: payload size

8 bits: Next Header type
8 bits: hop limit
128 bits: source address
128 bits: destination address

IPv6 doesn't have a checksum, since all upper level protocols need to implement their own anyway. Also, note that there aren't any options in the header. All IPv6 options are implemented in an Extension Header so that routers need not examine any part of the packet past the 40 byte header. IPv6 has no notion of fragments in headers – any necessary fragmentation must be done by the sending host, so hosts must implement MTU discovery if they want to use packets larger than the minimum size.

3 IPv6 Addresses

IPv6 addresses are 128 bits long, and like IPv4 addresses, specify an interface on a host.

IPv6 addresses are usually listed as a colon separated list of hex values such as 4321:0:1:2:3:4:567:89ab.

Any digits left out are taken as 0. there may be one "::" in an address which means to expand the address to include more than 32 contiguous 0 bits.

These are equivalent:
0:1:a::5fb
0000:0001:000a:0000:0000:0000:0000:05fb

4 Address Allocations

4.1 Aggretable Global Unicast Addresses

These are "normal" addresses, as they represent globally routable addresses. Addresses should be allocated by a provider (or by an exchange that serves several providers) The format is:

top 3 bits: 001
13 bits: TLA ID
8 bits: (reserved) 0x00
24 bits: NLA ID
16 bits: SLA ID
64 bits: interface ID

The TLA ID's (top-level aggregation ID) are meant to be allocated to tier-1 ISP's. (ISP's that form a complete mesh). The NLA (next-level aggregation) is meant to allow CIDR-style allocations and suballocations. The SLA (Site level level aggregation ID) is meant to be used for subnets within one location, either in a flat or aggregable allocation. The interface ID contains the MAC address of whatever link layer that interface is using. For example, the interface ID could contain a 48 bit ethernet address.

Example: 6bone addresses all begin with the prefix 3fee::/16 and are suballocated from there.

4.2 Embedded Addresses

4.2.1 IPv4-Compatible Addresses

IPv6 addresses that match 0:0:0:0:0:0::/96 contain an IPv4 address as the lowest 32 bits, and serve as automatic tunneling addresses for IPv4/IPv6 hosts. However, they allow none of the advantages of using IPv6, since there can only be 2^{32} IPv4-Compatible addresses, and they must be routed separately from normal unicast IPv6 addresses.

4.2.2 IPv4-Mapped IPv6 Addresses

A host with dual IPv4/IPv6 stacks can create IPv6 addresses for internal use from IPv4 addresses in the following manner: 0:0:0:0:0:FFFF:18.26.4.9

The host will then transform such an address into an IPv4 packet before sending it.

4.2.3 Multicast, Anycast Addresses, Site Local, Site Global, Loopback

There are address ranges allocated for Multicast, Anycast, Site Local, Site Global, and Loopback addresses. These all have analogs in IPv4 address space.

5 DNS modifications

In IPv4, a lookup asks for a type A record to return an address. To reverse resolve 18.26.4.9 you would lookup 9.4.26.18.IN-ADDR.ARPA for a PTR type record.

To get the ipv6 address associated with a name, ask for the AAAA record.
example:

```
% dig ipv6.linux-tech.com aaaa

; <<>> DiG 8.3 <<>> ipv6.linux-tech.com aaaa
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 5
;; QUERY SECTION:
;;      ipv6.linux-tech.com, type = AAAA, class = IN

;; ANSWER SECTION:
ipv6.linux-tech.com.    23h59m38s IN AAAA  3ffe:8114:1000::251

;; AUTHORITY SECTION:
linux-tech.com.         23h59m38s IN NS  taino.weblibre.org.
linux-tech.com.         23h59m38s IN NS  jade.weblibre.org.
linux-tech.com.         23h59m38s IN NS  onix.weblibre.org.
linux-tech.com.         23h59m38s IN NS  roca.weblibre.org.
linux-tech.com.         23h59m38s IN NS  lago.weblibre.org.

;; ADDITIONAL SECTION:
taino.weblibre.org.     1d23h59m23s IN A  66.123.163.229
jade.weblibre.org.      1d23h59m23s IN A  66.123.163.226
onix.weblibre.org.      23h59m27s IN A  66.123.163.227
roca.weblibre.org.      1d23h59m23s IN A  66.123.163.230
lago.weblibre.org.      1d23h59m23s IN A  66.123.163.228

;; Total query time: 1 msec
;; FROM: public-burning.lcs.mit.edu to SERVER: default -- 18.26.4.9
;; WHEN: Thu Sep 26 18:30:16 2002
;; MSG SIZE  sent: 37  rcvd: 253
```

To reverse resolve an address, you invert the digits and ask in the ip6.int domain:

```
% dig @taino.weblibre.org 1.5.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.4.1.1.8.e.f.f.3.ip6.int ptr

; <<>> DiG 8.3 <<>> @taino.weblibre.org 1.5.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.4.1.1.8.e.f.f.3.ip6.int ptr
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3
;; QUERY SECTION:
;;      1.5.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.4.1.1.8.e.f.f.3.ip6.int, type = PTR, class = IN

;; ANSWER SECTION:
1.5.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.4.1.1.8.e.f.f.3.ip6.int.  23h58m22s IN PTR  tunnel-251.ipng.nl.

;; AUTHORITY SECTION:
4.1.1.8.e.f.f.3.ip6.int.  23h58m22s IN NS  ns1.ipng.nl.
4.1.1.8.e.f.f.3.ip6.int.  23h58m22s IN NS  ns2.ipng.nl.
4.1.1.8.e.f.f.3.ip6.int.  23h58m22s IN NS  ns3.ipng.nl.

;; ADDITIONAL SECTION:
ns1.ipng.nl.           1d11h14m58s IN A  192.87.170.242
ns2.ipng.nl.           1d11h14m58s IN A  212.19.205.130
ns3.ipng.nl.           1d11h14m58s IN A  192.87.170.240

;; Total query time: 119 msec
;; FROM: public-burning.lcs.mit.edu to SERVER: taino.weblibre.org  66.123.163.229
;; WHEN: Thu Sep 26 18:47:45 2002
;; MSG SIZE  sent: 89  rcvd: 223
```

6 Application Interoperability

For the forseeable future, all IPv6 capable hosts will implement both IPv4 and IPv6, and probably have different IPv4 and IPv6 addresses as well. A machine with both routable IPv4 and IPv6 addresses can act as a server for either IPv4 or IPv6 clients, assuming that the network routes packets for both.

7 Network Interoperability

6Bone: overlay network consisting of IPv6-over IPv4 tunnels. Each operator sets up explicit tunnels for forwarding IPv6 packets embedded into IPv4 packets to hops, either at other 6bone sites, or to a IPv6 backbone.

6to4 relays: advertise 2002::/16 – don't need to construct tunnels for each destination, but do require relays to talk to hosts with IPv6 only addresses. Described in RFC3056.

8 For more info

The IPv6 test overlay network: `http://www.6bone.net/`

Pointers to many IPv6-related RFC's: `http://playground.sun.com/ipv6/specs/specifications.html`

All RFC's can be found here (and at many other sites as well) `http://www.rfc-editor.org`.