IPv6 Interface Class

Agenda

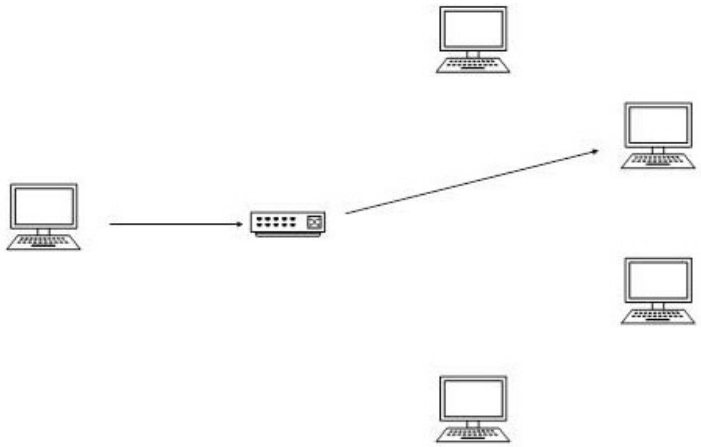
- IPv6
- Addressing Mode
- Addressing Format
- Addressing Model
- Header Format
- Extension Header
- IPv6 setupInterface Class

Why IPv6??

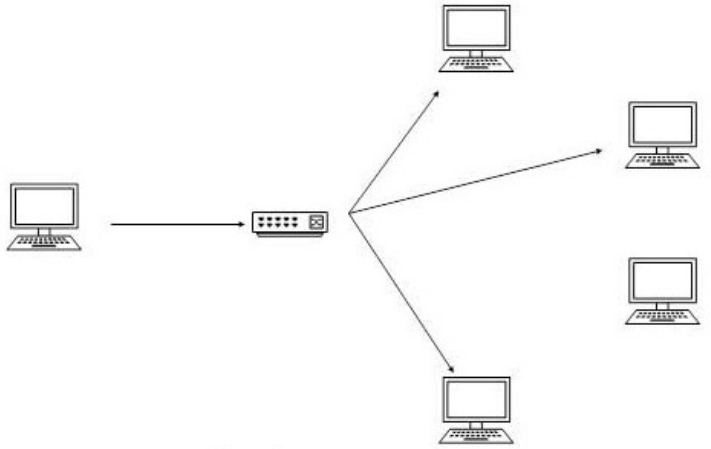
- Larger Address Space
- Simplified Header
- End-to-end Connectivity
- Auto-configuration
- Faster Forwarding/Routing
- IPSec
- No Broadcast
- Anycast Support
- Mobility
- Enhanced Priority Support
- Extensibility

Addressing Modes

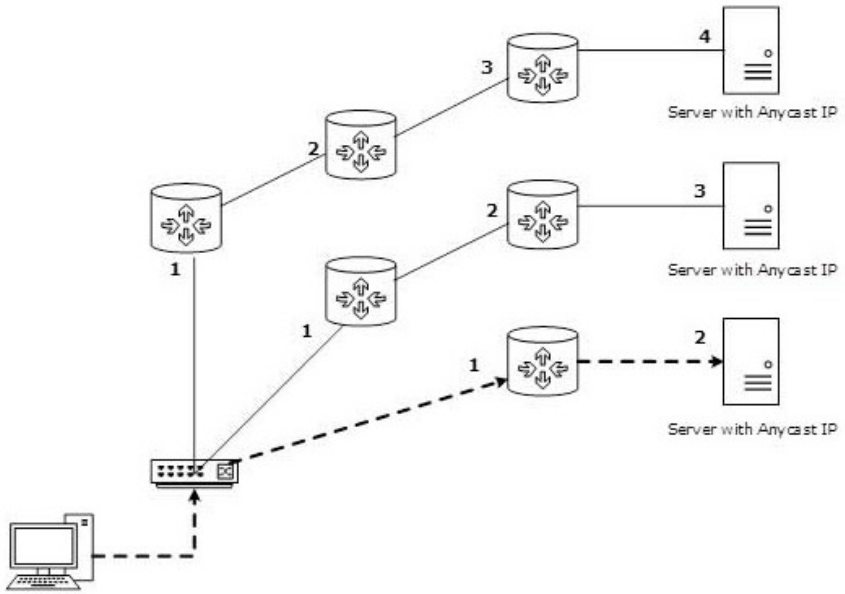
- Unicast



- Multicast



- Anycast



Anycast

- In this addressing mode, multiple interfaces (hosts) are assigned same Anycast IP address.
- Assume that all the Web Servers are assigned a single IPv6 Anycast IP Address.
- Now when a user from Europe wants to reach slscorp.com the DNS points to the server that is physically located in Europe itself.
- If a user from India tries to reach slscorp.com, the DNS will then point to the Web Server physically located in Asia.
- Nearest or Closest terms are used in terms of Routing Cost.

IPv6 Addressing

IPv4 32-bits

IPv6 128-bits

$2^{32} = 4,294,967,296$

$2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456$

$2^{128} = 2^{32} * 2^{96}$

2^{96} = 79,228,162,514,264,337,593,543,950,336 times the number of possible IPv4 Addresses (79 trillion trillion)

Addressing Format

Representation

- 16-bit hexadecimal numbers
- Numbers are separated by (:)
- Hex numbers are not case sensitive
- Abbreviations are possible

  Leading zeros in contiguous block could be represented by (::)

  Example:

  2001:0db8:0000:130F:0000:0000:087C:140B

  2001:0db8:0:130F::87C:140B

  Double colon only appears once in the address

IPv6 Address Representation

- Loopback address representation

  0:0:0:0:0:0:0:1=> ::1

  Same as 127.0.0.1 in IPv4

  Identifies self

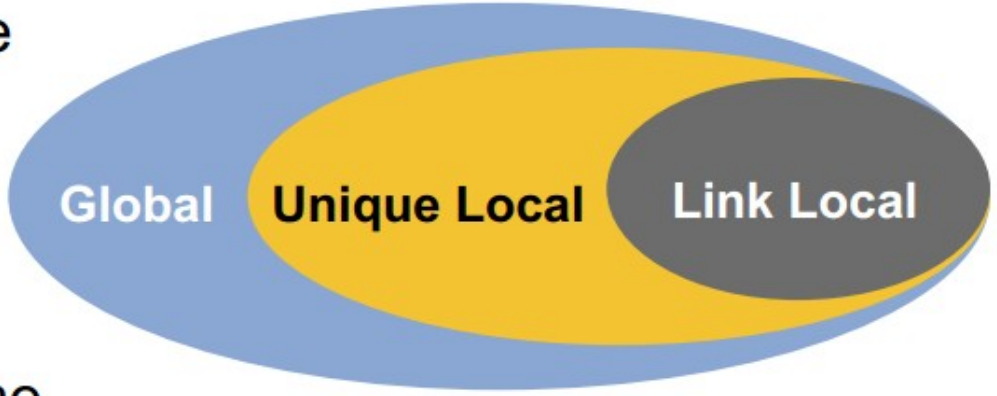
- Unspecified address representation

  0:0:0:0:0:0:0:0=> ::

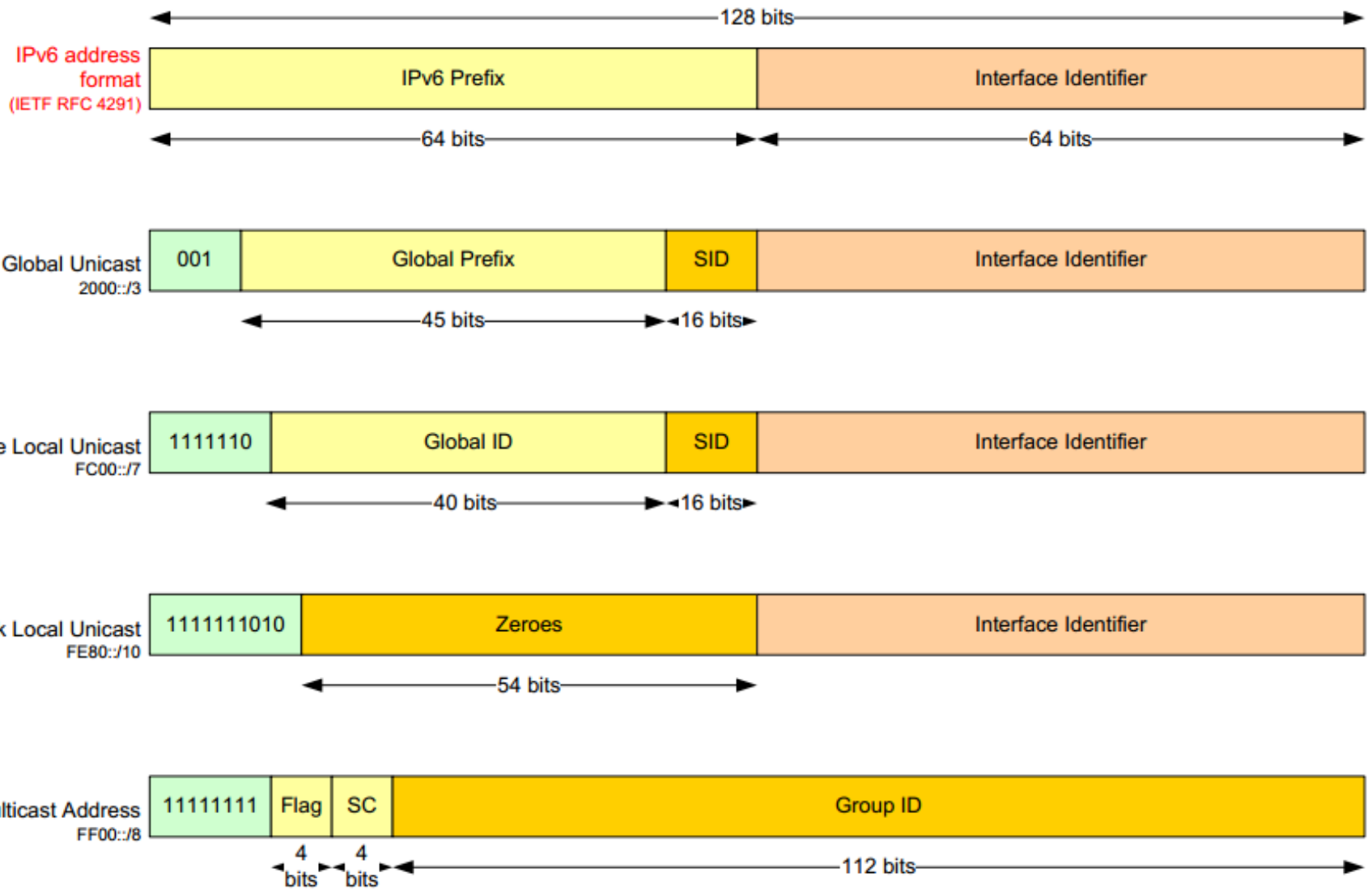
  Used as a placeholder when no address available

IPv6 Addressing Model

- Addresses are assigned to interfaces
  - Change from IPv4 mode
- Interface "expected" to have multiple addresses
- Addresses have scope
  - Link local
  - Unique local
  - Global
- Addresses have lifetime
  - Valid and preferred lifetime



Addressing


128 bits
IPv6 address format
(IETF RFC 4291)
IPv6 Prefix
Interface Identifier
64 bits
64 bits
Global Unicast
2000::/3
001
Global Prefix
SID
Interface Identifier
45 bits
16 bits
Unique Local Unicast
FC00::/7
1111110
Global ID
SID
Interface Identifier
40 bits
16 bits
Link Local Unicast
FE80::/10
1111111010
Zeroes
Interface Identifier
54 bits
Multicast Address
FF00::/8
11111111
Flag
SC
Group ID
4 bits
4 bits
112 bits

Addressing

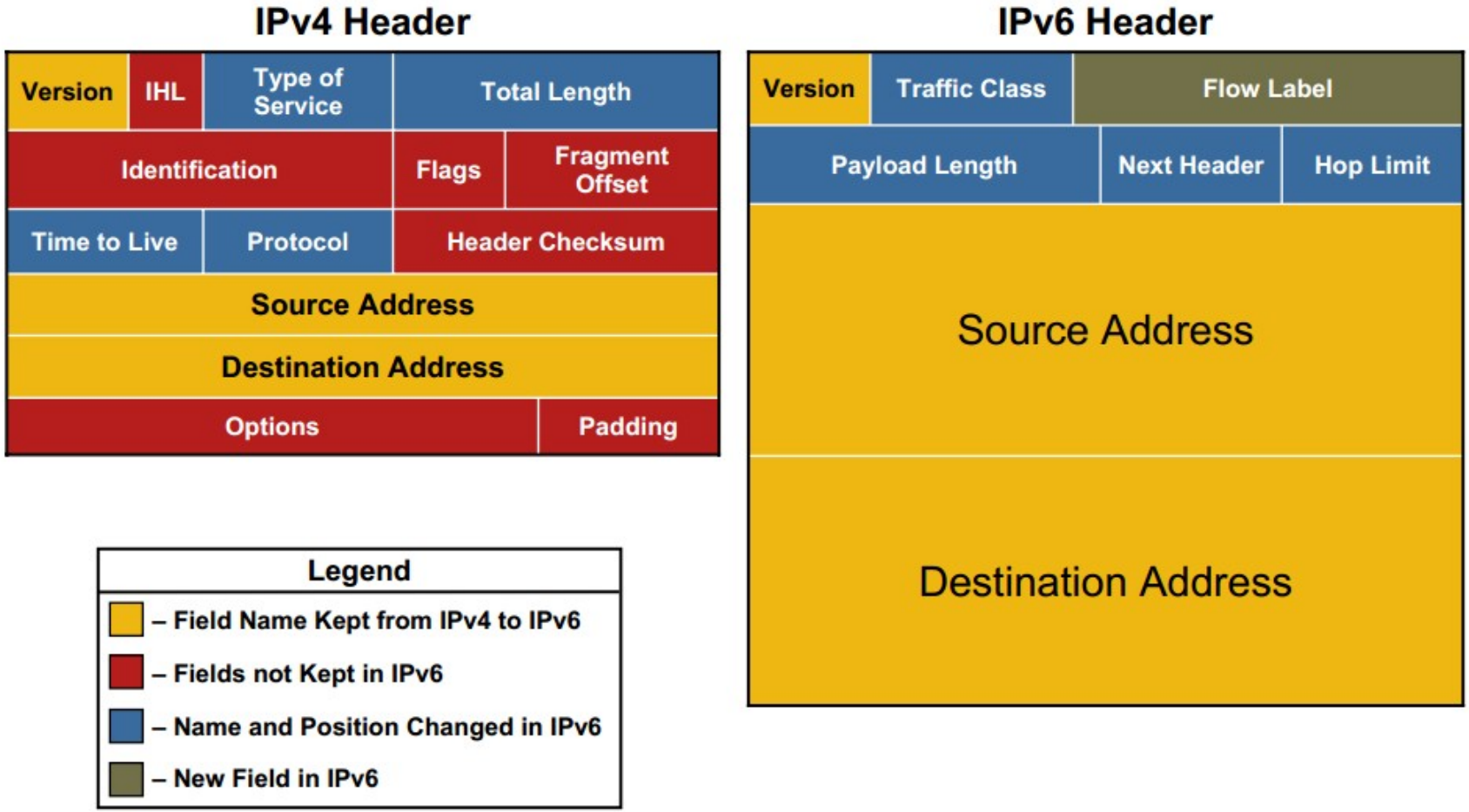
- Global prefix assigned by IANA
- Subnet ID (SID) allocated by the network administrator
- Interface Identifier either generated from the interface's MAC address or obtained from a DHCPv6 server, or assigned manually;

- Unique Local Unicast is an address only applicable to local network. This type of address is not routable outside the local network. The Global ID and the Subnet ID (SID) are allocated by the network administrator.

- Link Local Unicast is a unicast address allowed for a link local (without router). This type of address is not routable outside a local link

- Multicast is an address assigned to different devices of the network. Following the scope (SC) of the address, the multicast group may be either Interface-local, Link-local, Admin- local, Site-local, Organization-local or global.

Header Comparison Of IPv4 and IPv6



Version (4 bits) :- The constant 6 (bit sequence 0110).

Traffic Class (8 bits) :- These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router Known what services should be Provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN).

Flow Label (20 bits) :- This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information. This field helps avoid re-ordering of data packets. It is designed for streaming/real-time media.

Payload Length (16 bits) :-The size of the payload in octets, including any extension headers. The length is set to zero when a Hop-by-Hop extension header carries a Jumbo Payload option.

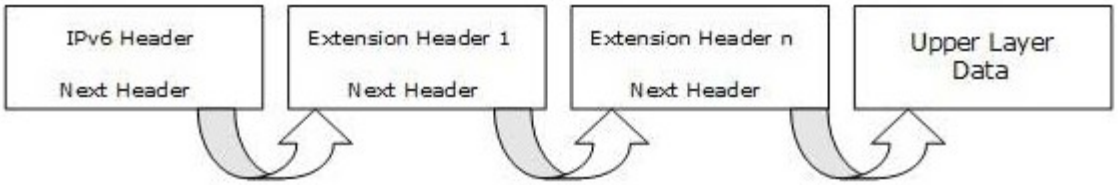
Next Header (8 bits) :- Specifies the type of the next header. This field usually specifies the transport layer protocol used by a packet's payload. When extension headers are present in the packet this field indicates which extension header follows.

Hop Limit (8 bits) :- Replaces the time to live field of IPv4. This value is decremented by one at each intermediate node visited by the packet. When the counter reaches 0 the packet is discarded.

Source Address (128 bits) :- The IPv6 address of the sending node.
Destination Address (128 bits) :- The IPv6 address of the destination node(s).

Extension Header

Extension Header	Next Header Value	Description
Hop-by-Hop Options header	0	read by all devices in transit network
Routing header	43	contains methods to support making routing decision
Fragment header	44	contains parameters of datagram fragmentation
Destination Options header	60	read by destination devices
Authentication header	51	information regarding authenticity
Encapsulating Security Payload header	50	encryption information



IPv6_setup Interace Class

IPv6 setup		**0...n**	**class_id = 48, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. DL_reference	(static)	octet-string				x + 0x08
3. address_config_mode	(static)	enum			0	x + 0x10
4. unicast_IPv6_addresses		array				x + 0x18
5. multicast_IPv6_addresses	(static)	array				x + 0x20
6. gateway_IPv6_addresses	(static)	array			0	x + 0x28
7. primary_DNS_address	(static)	octet-string			0	x + 0x30
8. secondary_DNS_address	(static)	octet-string			0	x + 0x38
9. traffic_class	(static)	unsigned	0	63	0	x + 0x40
10. neighbor_discovery_setup	(static)	array				x + 0x48
Specific methods		***m/o***				
1. add_IPv6_address (data)		*o*				x + 0x60
2. remove_IPv6_address (data)		*o*				x + 0x68

THANK YOU