Introduction to IPv6 Protocol Structure

Rocky Mountain IPv6 Summit

April 21 – 22, 2009

John Spence
spence@commandinformation.com
www.commandinformation.com

Command offers:

- Implementation consulting – specializing in provider deployments and Fed/DoD
- Training
  - Multi-day onsite classes, lab-heavy, self-contained (give us a conference room , power, and 20 engineers or developers and we will do the rest)
  - Building IPv6 Networks
  - IPv6 Application Development
  - Securing, Hacking, and Defending IPv6 Networks
  - IPv6 for Security Professionals

Flexible and Extensible

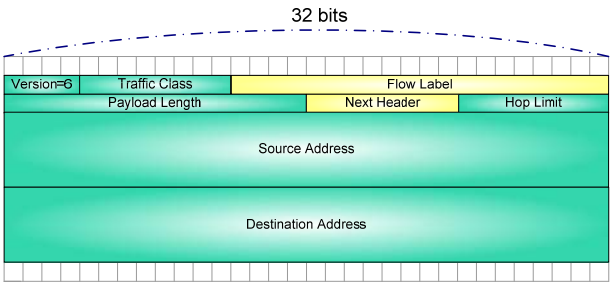
- IPv6 packet header structure and extension header structure provide the capability to perform additional L3 functions and support ongoing innovations in IP design
- Pushes more processing to edges, simplifies core routing
- Packet design provides support for
  - Partitioning of header elements into network centric (e.g. - “Hop-by-Hop” Options) and host centric (e.g. - “Destination Options”) categories
    - Without impacting the “cost” of forwarding these packets
    - Which also, in turn, enables more innovation in the IP layer
  - End-to-end functions like IPsec and peer-to-peer signaling
  - Network-based functions like QoS, and the potential for improved QoS handling in the future using the “flow” concept

IPv4 Header Structure Review

- 32-bit addressing field
- Header Checksum (error checking)
- Variable length “Options” field
- Fragmentation fields present
- Variable length header (20Bytes + Options)
- Aligned on 32-bit boundaries
- Fields in yellow not be present in IPv6 Header

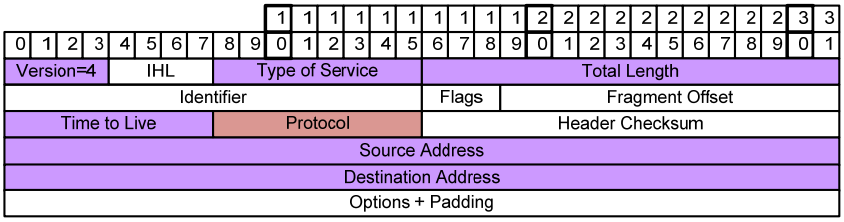
Features of IPv6 Header

- Fixed length = 40 bytes = no HL field = more efficient
- Fewer fields = more efficient
- No header error checking = more efficient
- Fragmentation fields removed = more efficient
- Streamlined, extensible (via extension header – coming up)
- Aligned on 64-bit boundaries (image drawn in 32 bit scale for ease of reading)
- Fixed 40-byte (Base) Header length

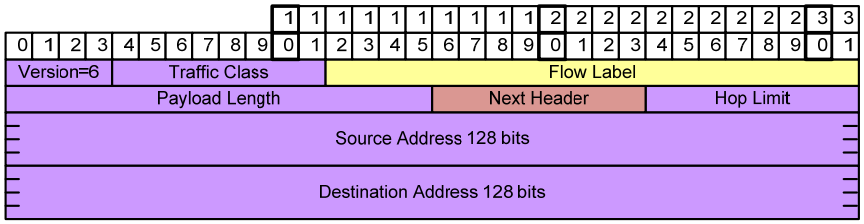


IPv4/IPv6 Header Comparison

IPv4 Header

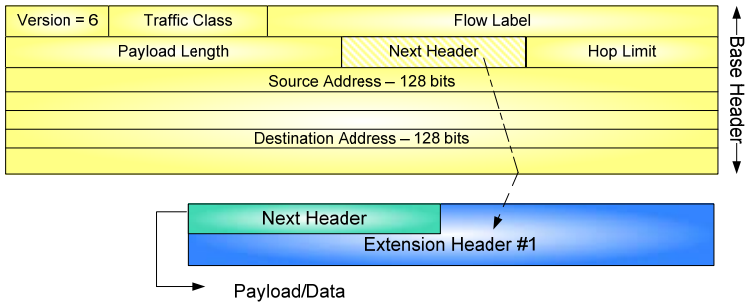


IPv6 Header



IPv6 Next Header Format

- The Next Header field indicates what type of header follows the current header
- Extension header information counted within “Payload Length”

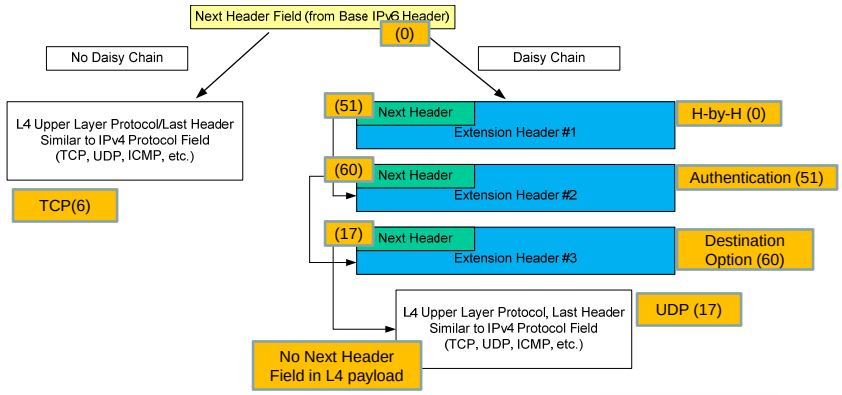


Extension Header Ordering

- Hop by Hop Options Header (value = 0) ← Must be first if present
- Destination Options Header (60) ← Has "length" field
  - Where all destinations specified in Routing Header also process these Destination Options
- Routing Header (43) ← Deprecated
- Fragment Header (44) ← No "length" field
- Authentication Header (51)
- Encapsulating Security Payload Header (50)
- Mobility Header (135)
- Destination Options Header (60)
  - Where no Routing Header is used
- ICMPv6, or L4 payload such as TCP, UDP (58, 6, 17, etc.)
- "No Next Header" (59)
  - *Note! Cannot skip unknown IPv6 Extension Header*

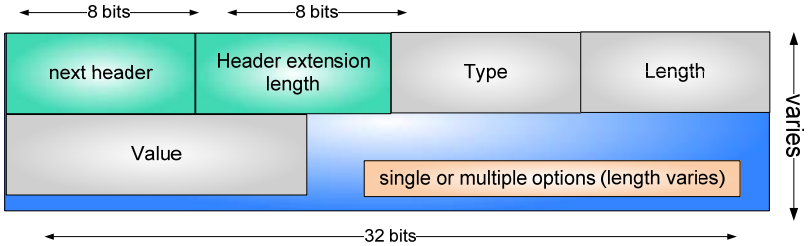
IPv6 Next Header Field in Detail

- The Next Header field can point to two categories, Upper Layer Protocol (TCP, UDP, ICMP, etc) or Extension Headers.
- The Upper Layer Protocol/Last Header, also known as the Protocol field in IPv4 can not be daisy chained.



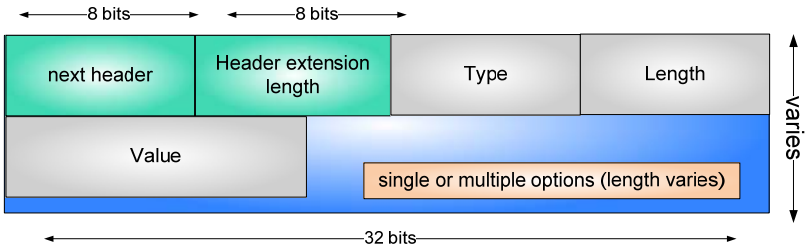
Hop-by-Hop Options Header

- Hop-by-Hop Options Header examined by all nodes in packet's path
- Header may contain multiple options
- Header may have padding
- Options encoded T-L-V (can be skipped, depending)
- Must be first or only extension header when present



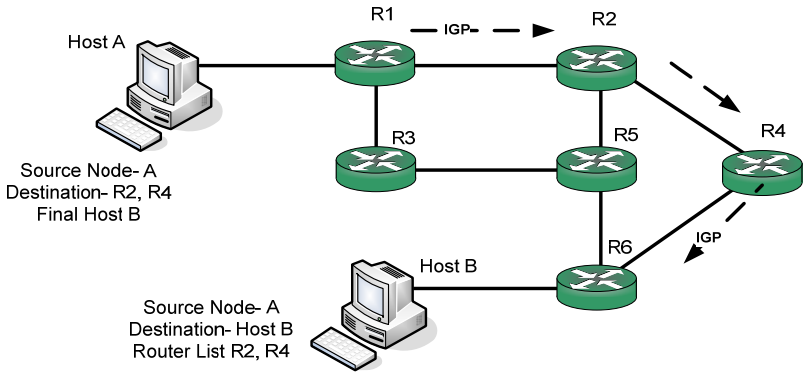
Destination Options Header

- Destination Options only examined by destination node
- Intermediate nodes do not examine D.O.
- T-L-V encoding; same layout as Hop-by-Hop
- Used by mobile IPv6 (as an example)



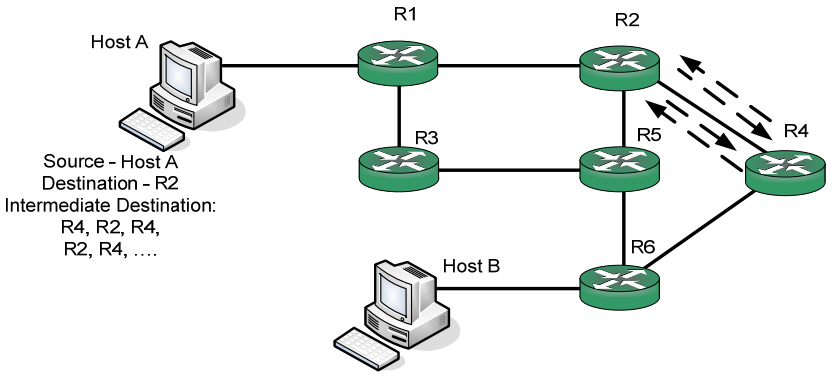
Routing Header - LSRR (Type 0)

- Forced routing by directing packets through intermediate hops



Routing Header (2)

- Type 0 Routing Header deprecated due to amplification attack.
- Block type 0 but not type 2 routing header if using MIPv6
- Type 2 routing header still valid - allows only one intermediate node, and must be used with MIPv6

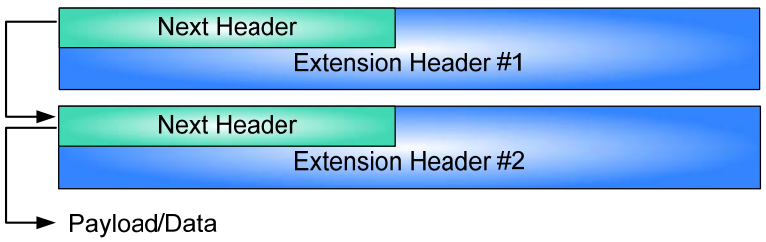


Fragment Header (1)

- Fragment header (44)
- Only source nodes fragment packets in IPv6
- Fragmentation and reassembly are host based only – no intermediate fragmentation available as in IPv4
- Needed for packets that exceed path MTU limits
- Offset, flags ("more fragments"), identification fields
  - Packets must have identical Source and Destination addresses
  - Packets must have same identification value
  - IPv6 uses 32-bit identification field (IPv4 uses 16-bit field)

IPsec Extension Headers

- Authentication header (51 – AH)
  - Provides source address authentication, packet integrity and anti-replay protection
- Encapsulating Security Payload (50 – ESP)
  - Provides confidentiality, source address authentication (optional), data integrity, and anti-replay protection

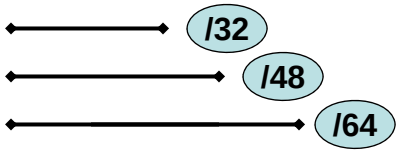


Summary

- IPv6 base header streamlined, simplified
- Daisy-chaining of extension headers provides flexibility, extensibility
- Options headers provide flexible capability for protocol extensions, split into two major types
  - Hop-by-Hop Options
  - Destination Options
- Packet with unknown extension header must be dropped
- Packet with unknown option may proceed, with processing for the unknown option skipped, as directed (or not) by option number encoding
- Host-based fragmentation only – no intermediate fragmentation

IPv6 Address – Format and Basics

2001:0DB8:00A7:8AC4:0234:7BFF:FE19:223C



- IPv6 address is 128 bits long
  - First 32 bits typically ISP (::/32)
  - First 48 bits typically Enterprise (::/48)
  - First 64 bits typically subnet (::/64)
  - Low 64 bits often includes interface MAC address
- Written in Hex, colon delineators into 16-bit “chunks”

The IPv6 Address Space

- IPv4 addresses 2^32 = 4,294,967,296
- IPv6 addresses 2^128 = 340,282,366,920,938,463,463,374,607,431,768,211,456 or
  - or, 340 undecillion (US) addresses
- 79,228,162,514,264,337,593,543,950,336 times more v6 addresses than v4
- If IP addresses weighed one gram each
  - IPv4 < 1/7th of the Empire State Building
  - IPv6 > 56 billion(US) earths

Writing Addresses

- The written format of an IPv6 address is ***<address>/<prefix length>***. Example:

  2001:0DB8:0049:0000:AB00:0000:0000:0102/64

- /64 in the above example is the number of leftmost bits in the address that constitutes the prefix
- It is common for addresses to contain many 0 (zero) bits

Drop Leading Zeros

2001:0DB8:0400::/48	(original)
2001:DB8:400::/48	(correct)
2001:DB8:04::/48	(wrong! – removed trailing zeros)
(invented network 2001:DB8:0004::/48)	
2001:0DB8:0049:0000:AB00:0000:0000:0102/64	(original)
2001:DB8:49:0:AB00:0:0:102/64	(correct)

- Address can be written in more concise format by removing **leading** zeros in any chunk
- Node fills out addresses before using – terse format is just for readability

Combine All-Zero "Chunks"

2001:0DB8:0049:0000:AB00:0000:0000:0102/64	(original)
2001:DB8:49:0:AB00:0:0:102/64	(correct – not yet fully compressed)
2001:DB8:49:0:AB00::102/64	(correct – fully compressed)
2001:DB8:49::AB00::102/64	(wrong!)
(cannot have two sets of double-colons in address)	

- Consecutive all-zero chunks can be condensed with "double-colon" notation
- Can only use it once in an address
  - Last line of example – no good
    - 2001:DB8:49:0:AB00:0:0:102 ??
    - 2001:DB8:49:0:0:AB00:0:102 ??

IPv6 Address Types

- Unicast
  - One-to-one communication
- Multicast
  - One-to-many communication
  - Scope field better defines who receives data
  - Fundamental for neighbor discovery, router advertisements and other critical IPv6 mechanisms
- Anycast
  - One to one-of-many communication
  - Communication between a single sender and the (one) nearest of several possible (many) receivers in a group
  - Quickly and easily locates the closest server that has the information being requested

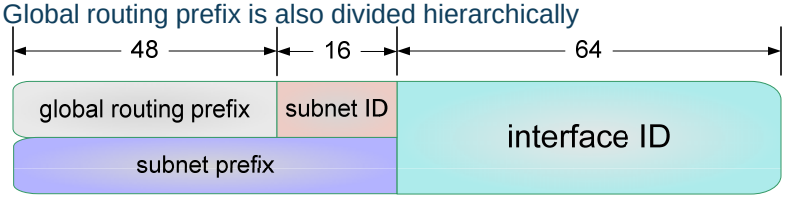
Identifying Addresses

- These are the address types and their binary representations:

Address Type	Binary Prefix	IPv6 Notation
Unspecified	`00 … 0 (128 bits)`	`::/128`
Loopback	`00 … 1 (128 bits)`	`::1/128`
Link-local unicast	`1111111010`	`FE80::/10`
Unique Local unicast	`1111110`	`FC00::/7`
Site-local unicast (deprecated)	*`1111111011`*	*`FEC0::/10`*
Multicast	`11111111`	`FF00::/8`
Global unicast	(everything else)	

Global Unicast Addresses

- Most common address type is global unicast
- Interface ID is usually 64 bits, leaving 64 bits for subnet prefix
- Subnet prefix is composed of
  - Global routing prefix – assigned to a site by provider
  - Subnet ID – identifies link within site
- Global routing prefix is also divided hierarchically

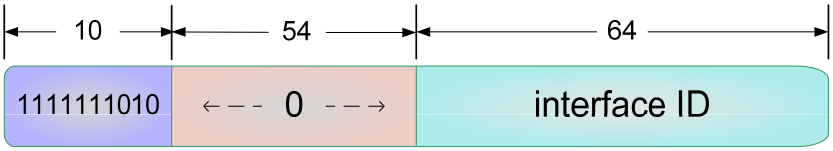


IID looks random

Examples: `2001:DB8:67EA:FE67:810A:789E:AE:78B2`
`2001:DB8:67EA:B::5`

IID looks chosen

Link-local Addresses



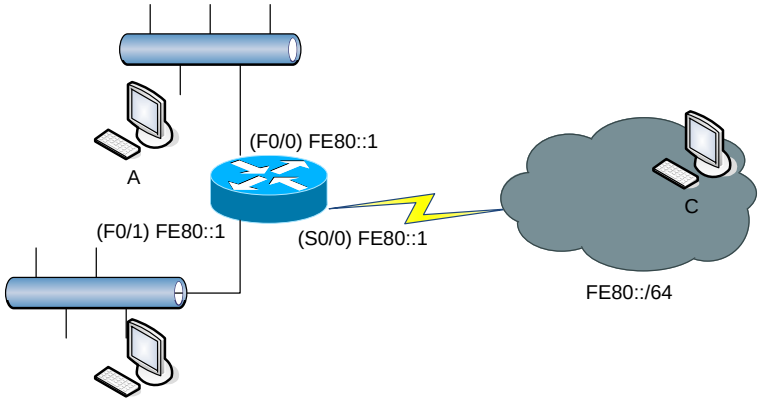
- Link-local addresses are only valid on a single link, or subnet
- Always begin with the prefix “FE80::/10”, then contain 54 bits of zeros, followed buy the 64-bit Interface ID
- Can be automatically generated or manually configured on an interface

IID embeds MAC Address 01-23-45-67-89-0A

Examples: `FE80::323:45FF:FE67:890A`
`FE80::200`

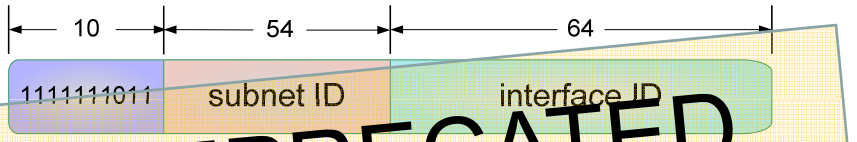
IID looks chosen

Link-local Address Concept



- This is a valid configuration for link-local addressing
- Note hosts A nor B nor C can reach each other

Site-local Addresses (Deprecated)



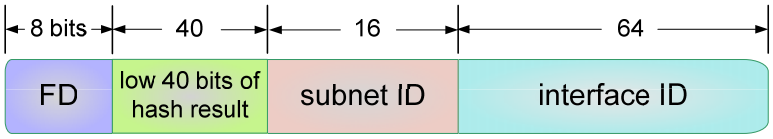
DEPRECATED

- Site-local addresses valid within a site
- Not unlike RFC 1918 v4 addresses (10.0.0.0/8)
- Format as shown – note huge subnetting space (54 bits)
- **Site-local addresses have been deprecated in favor of Unique Local**

- **Problem: What if organizations mostly choose FEC0:0:0::/48?**
- **Big address space, all clustered in same /48 – like RFC1918 10 Net**

Examples: `FEC0:0:0:0::323:45FF:FE67:890A`
`FEC0:0:0:1::2`

Unique Local Addresses



- Site-scoped prefix
- Unique local addressing creates a non-routable prefix for use within an organization that is statistically likely to be globally unique
- Not routable on Internet; routable within organization
  - Or between organizations over a private link
- Described in RFC 4193
- Registered version (FC00::/8) may be defined later

For example: FD3A:84E2:4FE2::/48 is Command Information's unique local address.

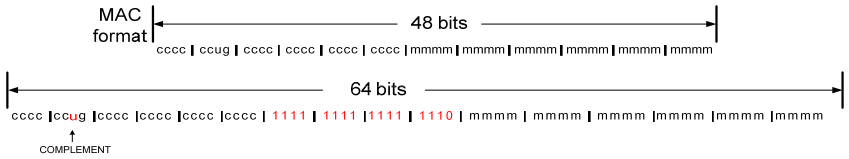
40 bits randomized

64-bit Interface Identifiers

- Interface IDs identify interfaces on a link
  - They must be unique on the link
  - They need not be unique across multiple links
    - A single node on multiple links can use the same interface ID on all links
- Interface IDs may be unique across a wider scope
  - In fact, some may be globally unique – such as where IID is based on a globally-unique MAC address
- Some IIDs are reserved
  - Example: all-zeros in IID is subnet-router anycast IID
  - Example: certain high IIDs are reserved subnet anycast IID

EUI-64 Construction Rules - Construction

- Current Ethernet cards have 48-bit MAC
  - Insert “FF-FE” (16 bits) between OUI and serial number
- Convert that to a Modified EUI-64 interface ID
  - Complement the “universal/local” bit



EUI-64 Construction Rules - Example

48 bit MAC address: 00-23-45-67-89-0A

- Result is 64-bit Modified EUI-64 interface ID that is globally unique
  - Note: This is an interface ID – not an IPv6 address

02-23-45-FF-FE-67-89-0A

Bit seven complement

- Corresponding link-local address:

FE80::0223:45FF:FE67:890A

IPv6 Privacy/Temporary Addresses (IIDs, really)

- The IPv6 address low 64-bit IID used by SLAAC does not change over time since it uses the IEEE MAC identifier of the node's NIC
- IPv6 *autoconfigured* address can be tracked over time
  - John at work = 2001:DB8:4:5:323:45FF:FE67:890A
  - John in Tokyo = 2609:12:AE:B675:323:45FF:FE67:890A
  - Geo-location techniques make it easy to track device location
- Privacy Addresses randomize an IPv6 address IID so that there is no fixed EIU-64 identifier over time to allow a device to be tracked despite the (possibly) changing /64 prefix
  - John at work = 2001:DB8:4:5:412:650A:8BB2:BEA6
  - John in Tokyo = 2609:12:AE:B675:659:3481:27BC:17EB
  - No way to correlate two addresses to one device
- Downside: privacy addresses makes it hard for an administrator to track systems or debug problems
  - Deploy in unmanaged environment – home environments
  - On by default in Vista, for example

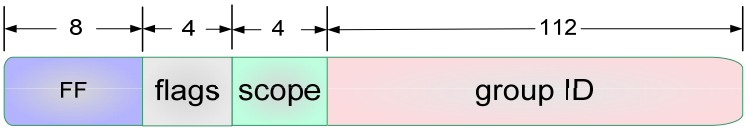
Random Interface IDs *replace* EUI-64 IIDs

- Microsoft introduced a different randomized IID for IPv6 addresses
- Note that IPv6 Privacy Addresses are generated *in addition to* autoconfigured EUI-64 format
- So, even if using "Privacy Addresses", which are intended for anonymous clients to use to connect to published servers, autoconfigured addresses with the embedded MAC are initialized in the interfaces and "valid", even if not "preferred"
- Microsoft invented randomized IIDs to make it harder to scan an IPv6 network looking for certain predictable IIDs
- Note Vista machine has no EUI-64 IIDs – not even in link-local

```
Wireless LAN adapter IntelProWireless:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:470:e863:1:b580:75d:6bbb:5963
   Temporary IPv6 Address. . . . . . : 2001:470:e863:1:9930:887a:3239:430a
   Link-local IPv6 Address . . . . . : fe80::b580:75d:6bbb:5963%10
   IPv4 Address. . . . . . . . . . . : 10.20.16.67
   Subnet Mask . . . . . . . . . . . : 255.255.255.128
   Default Gateway . . . . . . . . . : fe80::216:c8ff:fe6b:8c61%10
                                       10.20.16.1
```

Multicast Address Format



0000 = permanent (IANA assigned)

0001 = temporary (locally/randomly assigned)

- Multicast addresses can be listened to by multiple nodes at once – even on the same link
- Always begin with "FF"
- The last 112 bits are the multicast group ID
- Not all flags shown here

Multicast Address Scoping

- IPv6 has powerful scoping rules
- Link-local multicasting, for example, is used extensively in IPv6
- Permanent multicast assignments can be "of any scope"
- 16 scopes total – not all shown in table

Hex value	Scope
0x0	Reserved
0x1	Loopback
0x2	link-local
0x5	site-local
0x8	organization-local
0x9	unassigned
0xE	global

Example: Temporary site-local scoped multicast
FF15:200:300::AAAA

Common Multicast Addresses

FF01::1	all nodes multicast
FF02::1	

FF01::2	all routers multicast
FF02::2	
FF05::2	

FF02::9	all RIP routers multicast

FF02::1:FFxx:xxxx	solicited node multicast

- Note that all-nodes-multicast is functionally equivalent to broadcast

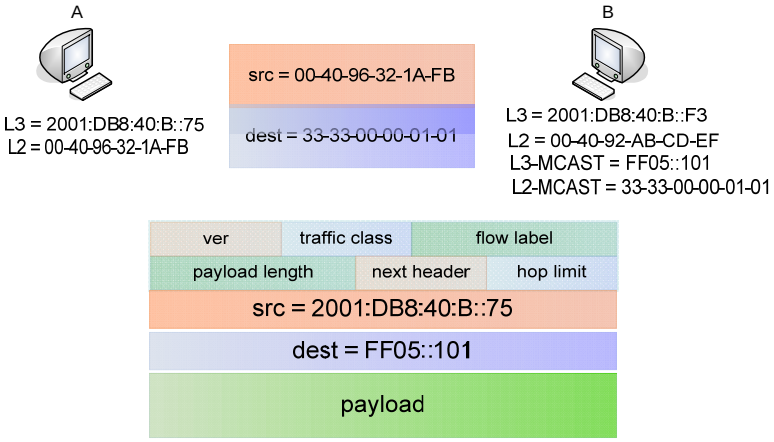
Multicast Address Mapping on Ethernet

- Into what multicast MAC address should a multicast IPv6 packet be placed?
- Mechanism to "map" L3 to L2 shown below – "33-33" is assigned by IANA for this purpose
- Append last 32-bits of L3 (IPv6) multicast address
  - Very much like IPv4 multicast mapping and low 23-bits

IPv6 multicast address
FF02:1234:5678:90AB:CDEF:1234:*5678:90AB*

Layer 2 multicast MAC **33-33-*56-78-90-AB***

IPv6 Multicast Mapping Example



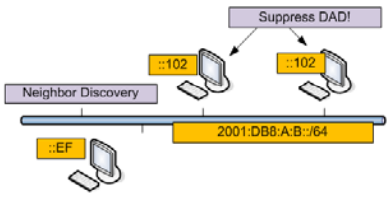
- L2 multicast destination based on L3 multicast destination

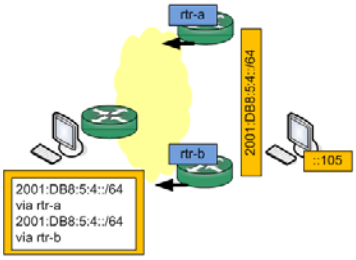
Anycast Addresses

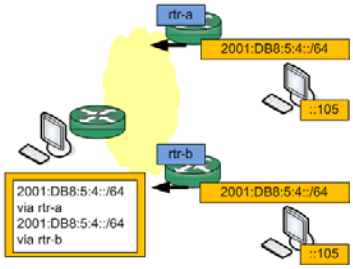
- Anycast addresses are used to reach a “nearest” instance of a given address, where multiple nodes have been assigned the same (anycast) address
- Drawn from the unicast address space – no special format – not immediately recognizable
- Must tell an interface at configuration time if you are giving it an anycast address

Anycast can be implemented in the LAN or WAN

- At right, anycast on the LAN
- Below right, Internet anycast
- Best example of production Anycast is DNS servers







Required Addresses

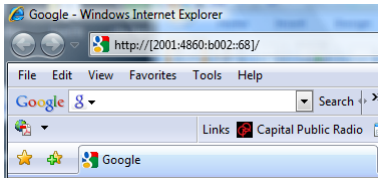
- Interfaces will have many addresses – host

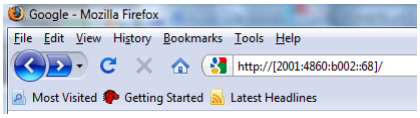
Address type	
Link-local	Required for each interface
Additional Unicast and Anycast	Optional (Manually or automatically configured)
Loopback	Required
All-Nodes Multicast	Required
Solicited-Node Multicast	Required for each of its unicast and anycast addresses
Multicast (Application based)	Optional (Of all other groups to which the node belongs)

- Router has additional – including “all routers multicast”

IPv6 Address in URL Must be in Square Brackets

- IPv6 address – with colon notation – unfriendly to traditional URL notation
- http://10.10.10.5:8080 ":8080" means "port 8080"
- RFC 3986 describes literal URL format for IPv6
- Enclose IPv6 address in [ ] (square brackets)
- Examples → http://[2001:4860:B002::68]:8080
- Caution: WinXP IE does not support literal URL format!



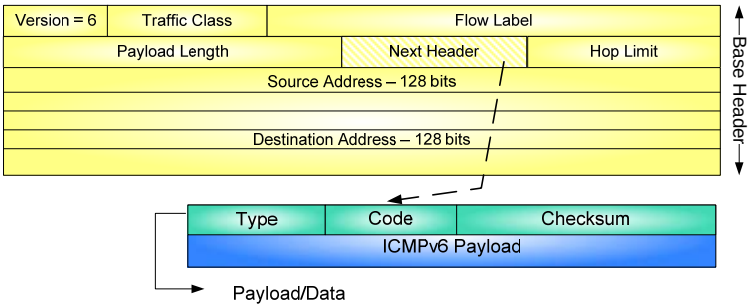


ICMPv6 and Neighbor Discovery

- ICMPv6 is a critical protocol that provides informational and network error messages.
- Neighbor Discovery (RFC4861) is a key factor in IPv6 for address auto configuration, host location and more. It uses certain ICMPv6 messages to achieve this.

ICMP Next Header Format

- ICMPv6 is one of the “Next Header” values (58)
- ICMPv6 similar to ICMPv4 in that it provides
  - Diagnostic informational messages
  - Error reporting messages



ICMPv6 Supports Familiar ICMPv4 Functions

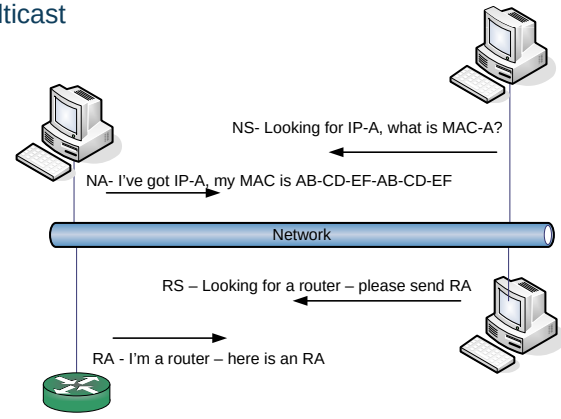
- Router redirect
- Destination unreachable
- Packet too big
- Time exceeded
- Parameter problem
- Echo request/reply (ping)

ICMPv6 Additional Messages

- New messages were added to ICMPv6 to support the Neighbor Discovery (ND) Protocol
  - Determines the link-layer address of neighbors on same local link – this is **Neighbor Discovery** – *replaces IPv4's ARP*
    - Duplicate Address Detection uses these messages as well
  - Router Discovery, find routers and get information from them to – among other things – perform address autoconfiguration
  - Neighbor Unreachability Detection (NUD) actively tracks reachability between active neighbors

Neighbor Discovery Messages

- Neighbor Solicitation – multicast not broadcast
- Neighbor Advertisement
- Router Solicitation - multicast
- Router Advertisement



Neighbor Solicitation Address Formation

Destination Node's Unicast Address

last 24 bits of the Unicast address

2001:DB8::1234:5678:9ABC

Neighbor Solicitation Multicast Format

FF02::1:FFxx:xxxx

Neighbor Solicitation Multicast Address

last 32 bits of the NS Multicast address

FF02::1:FF78:9ABC

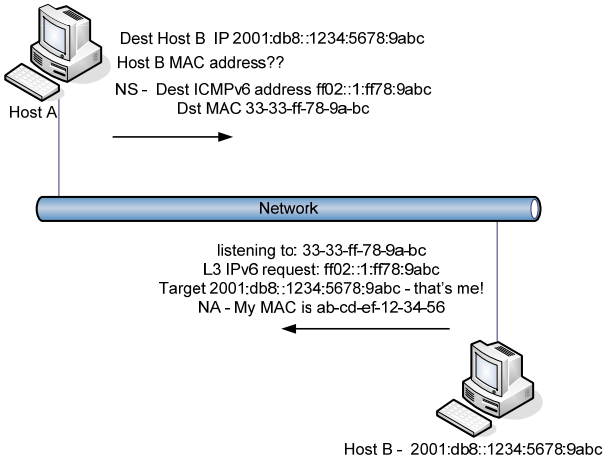
NS Multicast Layer Two Multicast MAC Address

pre-pended with "33-33" pad

33-33-FF-78-9A-BC

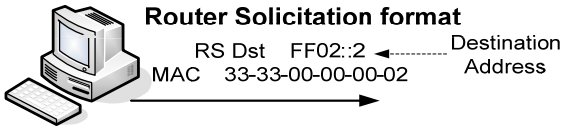
Neighbor Solicitation Process

- Host A needs to fill neighbor cache with Node B's MAC address
- Likely only B will receive NS because of sol-node multicast process
- NS/NA process also used for Duplicate Address Detection ("DAD") test – make sure address is unique before using it



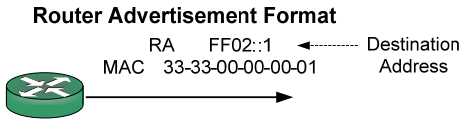
Router Solicitation (RS)

- A node can request a Router Solicitation on-demand
- When interface is initialized, rather than wait for periodic RA, interface will may send RS
- Sent to “all routers” multicast address



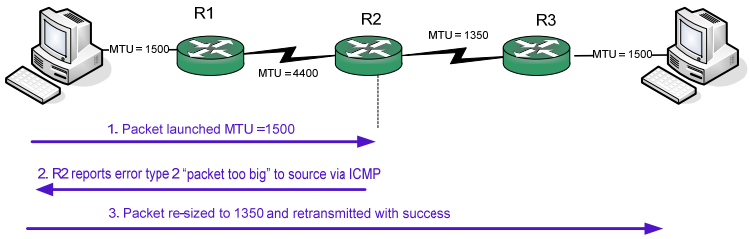
Router Advertisement (RA)

- Router Advertisement (RA) provides prefix information and other useful parameters to link-local nodes
  - Sent periodically and on-demand
  - RA includes “router lifetime” to indicate default router candidate
  - RA includes valid & preferred lifetime values for prefixes
  - RA can be configured to tell node to use DHCP
  - RA can also carry other information, such as Hop Limit
  - RA can carry default router preference and more specific routes
  - RA can carry option specifying recursive DNS server addresses



Path MTU Discovery

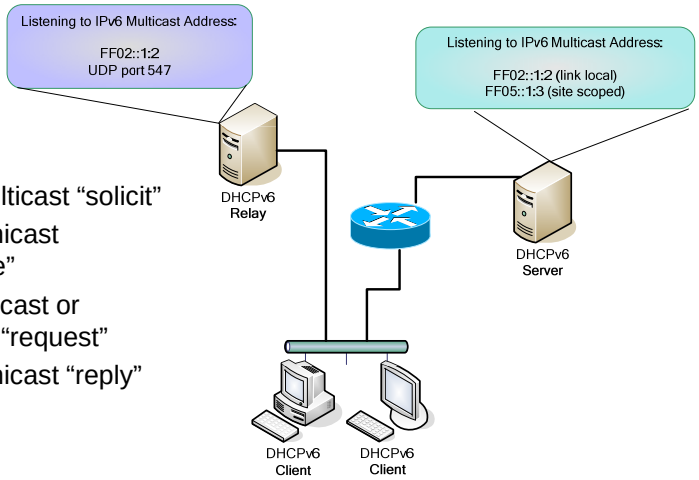
- PMTUD (RFC1981) uses ICMPv6 “packet too big” error message
- It is strongly recommended for IPv6 stacks to support MTU discovery, applications may or may not use it *(may choose to simply default to MTU of 1280-byte)*



DHCPv6

- DHCPv6 “stateful” addressing mechanism for IPv6
- Very similar to DHCP for IPv4
- Interesting features:
  - “stateful” configuration used for address assignment and setting other parameters
  - “stateless” configuration does not provide addresses – only “other” configuration parameters (perhaps SNTP server address)
  - DHCPv6-PD provides for delegation of entire prefix – not just single address or parameter
  - Currently, no DHCPv6 option exists to set a hosts “default router” – must be done from Neighbor Discover RA (but, IETF draft in progress to add capability to DHCPv6)

DHCPv6 Deployment Example



- Client multicast “solicit”
- Server unicast “advertise”
- Client unicast or multicast “request”
- Server unicast “reply”

- The example shown uses all three DHCPv6 components – DHCP client, relay, and server.

Reachable by IPv4, IPv6, or Either?

- Authoritative DNS entries control IP transit choice
- www.ietf.org reachable by either IPv4 or IPv6 – dual-stack
- www.google.com is IPv4-only service
- ipv6.google.com is IPv6-only service

```
C:\Users\jesO>host -v -t any www.ietf.org
Trying "www.ietf.org"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 777
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;www.ietf.org.                  IN      ANY

;; ANSWER SECTION:
www.ietf.org.           1800    IN      A       64.170.98.32
www.ietf.org.           1800    IN      AAAA    2001:1890:1112:1::20
www.ietf.org.           1800    IN      MX      0 mail.ietf.org.
```

```
C:\Users\jesO>host -v -t any www.google.com
Trying "www.google.com"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 369
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com.                        IN      ANY

;; ANSWER SECTION:
www.google.com.         0       IN      CNAME   www.l.google.com.
www.l.google.com.       244     IN      A       209.85.225.147
www.l.google.com.       244     IN      A       209.85.225.104
www.l.google.com.       244     IN      A       209.85.225.99
www.l.google.com.       244     IN      A       209.85.225.103
```

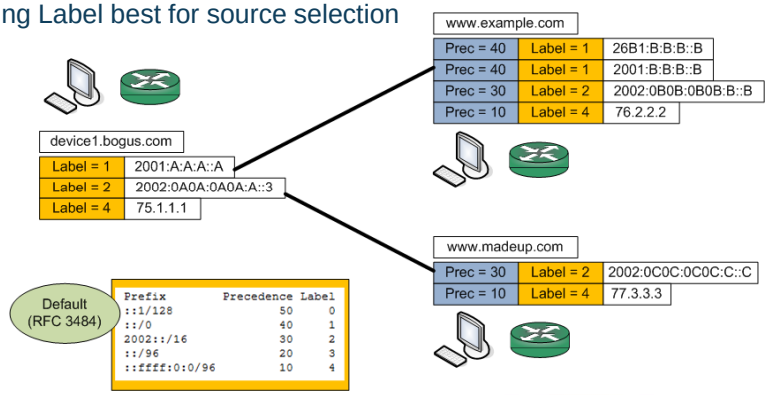
```
C:\Users\jesO>host -v -t any ipv6.google.com
Trying "ipv6.google.com"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1414
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ipv6.google.com.               IN      ANY

;; ANSWER SECTION:
ipv6.google.com.        0       IN      CNAME   ipv6.l.google.com.
ipv6.l.google.com.      300     IN      AAAA    2001:4860:b002::68
```

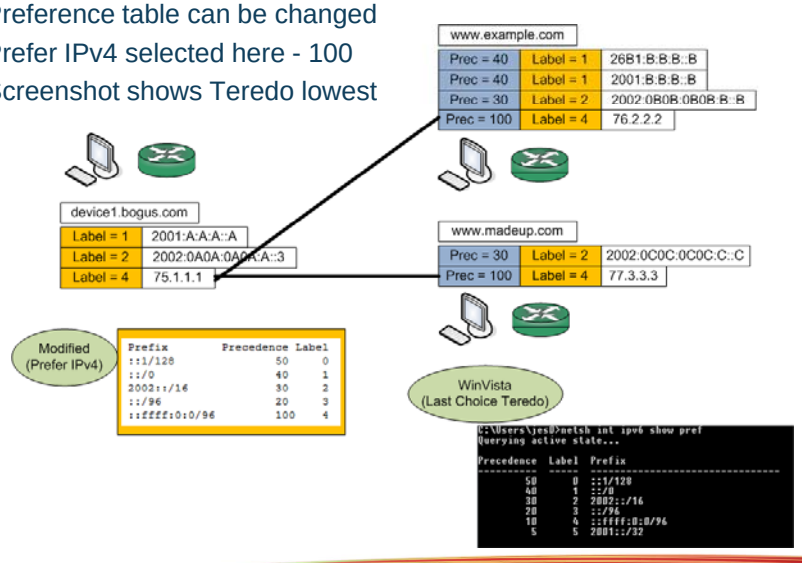
Default Address Selection – 18 rules

- IPv6 allows multiple addresses per interface
- Prefer IPv6 native, then other IPv6, then IPv4, longest match
- Policy table configurable
- Higher Precedence better for destination selection
- Matching Label best for source selection



Default Address Selection – Configurable

- Preference table can be changed
- Prefer IPv4 selected here - 100
- Screenshot shows Teredo lowest



Done … whew …

Thanks. Keep in touch.

John Spence
spence@commandinformation.com