HO CHI MINH CITY UNIVERSITY OF SCIENCE
FACULTY OF ELECTRONICS AND TELECOMMUNICATIONS
DEPARTMENT OF TELECOMMUNICATIONS AND NETWORKS

COURSE

COMPUTER NETWORKS

Chapter 05

IP VERSION 6

Reference: Peter L Dordal, "An Introduction to Computer Networks," Jul 26, 2019

Lecturer: Nguyen Viet Ha, Ph.D.   Email: nvha@fetel.hcmus.edu.vn

IPv6 – Why We Need More Address Space

IPv4 Address Allocation

1993

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Allocated
Unavailable
Available

16,777,216 addresses

IPv6 – Why We Need More Address Space

IPv4 Address Allocation

2000

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Allocated
Unavailable
Available

16,777,216 addresses

IPv6 – Why We Need More Address Space

IPv4 Address Allocation

2007

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Allocated
Unavailable
Available

16,777,216 addresses

IPv6 – Why We Need More Address Space

IPv4 Address Allocation

2009

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Allocated

Unavailable

Available

16,777,216 addresses

IPv6 – Why We Need More Address Space

IPv4 Address Allocation

2010

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Allocated

Unavailable

Available

16,777,216 addresses

IPv6 – Why We Need More Address Space

IPv4 Address Allocation

February 2011

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Allocated

Unavailable

Available

16,777,216 addresses

IPv6 – Why We Need More Address Space

❖ **Address Availability:**

- ➢ **IPv4:** 4 octets - 32 bits
  - ○ 2^32 *or* 4,294,467,295 IP Addresses.
- ➢ **IPv6:** 16 octets - 128 bits
  - ○ 2^128 ≈ 3.4 x 10^38 *or*

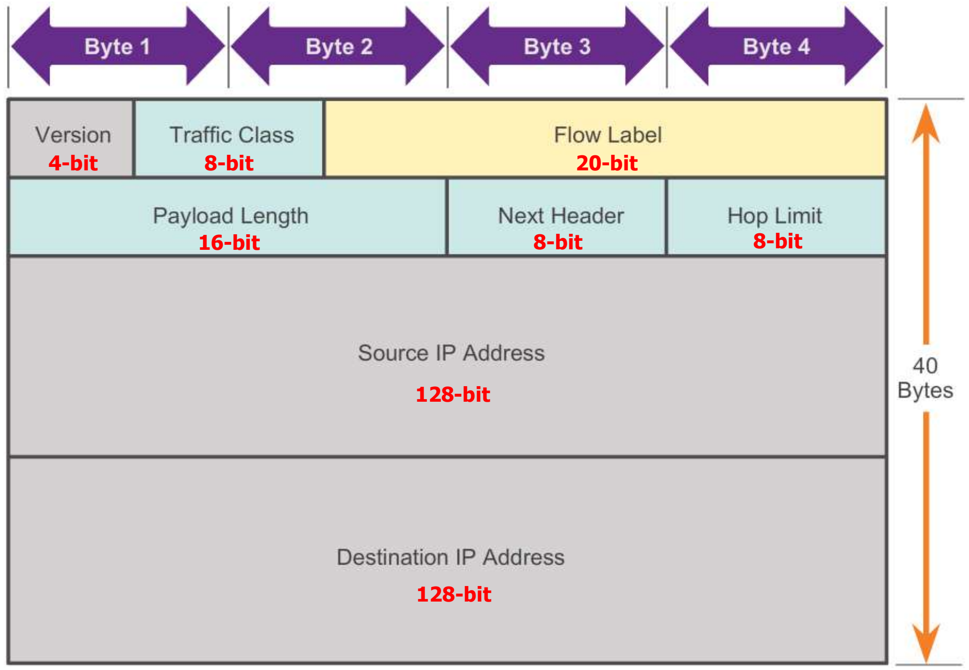
340,282,366,920,938,463,463,374,607,431,768,211,456

(340 undecillion) IP Addresses.

Number Name	Scientific Notation	Number of Zeros
1 Thousand	10^3	1,000
1 Million	10^6	1,000,000
1 Billion	10^9	1,000,000,000
1 Trillion	10^12	1,000,000,000,000
1 Quadrillion	10^15	1,000,000,000,000,000
1 Quintillion	10^18	1,000,000,000,000,000,000
1 Sextillion	10^21	1,000,000,000,000,000,000,000
1 Septillion	10^24	1,000,000,000,000,000,000,000,000
1 Octillion	10^27	1,000,000,000,000,000,000,000,000,000
1 Nonillion	10^30	1,000,000,000,000,000,000,000,000,000,000
1 Decillion	10^33	1,000,000,000,000,000,000,000,000,000,000,000
1 Undecillion	10^36	1,000,000,000,000,000,000,000,000,000,000,000,000

1 The IPv6 Header

1. The IPv6 Header



1. The IPv6 Header

- ❖ Fixed header (40 bytes)
  - ➢ No support for fragmentation
  - ➢ No header checksum
  - ➢ No option fields.

    - ○ **Extension headers** has been introduced to support some of these as options. (latter in Section 5)

1. The IPv6 Header

- ❖ **Version:** 4-bit, identifying the IP packet version (0110).
- ❖ **Traffic Class:** 8-bit, is equivalent to the IPv4 'Differentiated Services' field
- ❖ **Payload Length:** 16-bit, is equivalent to the IPv4 'Total Length' field.
- ❖ **Next Header:** 8-bit, is equivalent to the IPv4 'Protocol' field.
- ❖ **Hop Limit**: 8-bit, replaces the IPv4 'TTL' field.
- ❖ **Source Address:** 128-bit, identifies the IPv6 address of the sending host.
- ❖ **Destination Address:** 128-bit, identifies the IPv6 address of the receiving host.

1. The IPv6 Header

❖ **Flow Label:** 20-bit, be used by a source to label sequences of packets for which it requests **special handling** by the IPv6 routers, such as:
- Non-default quality of service
- "real-time" service.

➢ A flow, as the term is used here, is one-way; the return traffic belongs to a different flow.

➢ All packets belonging to the same flow should have the same Routing Extension header.

2 IPv6 Addresses

2. IPv6 Addresses

❖ **Hexadecimal** is a base sixteen system

❖ Base 16 numbering system uses the numbers 0 to 9 and the letters A to F

❖ Four bits (half of a byte) can be represented with a single hexadecimal value

Hexadecimal	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

2. IPv6 Addresses

❖ 128 bits in length and written as a string of hexadecimal values

❖ 32 hexadecimal values = IPv6 address

2001:0DB8:0000:1111:0000:0000:0000:0200
FE80:0000:0000:0000:0123:4567:89AB:CDEF

❖ **Hextet** used to refer to a segment of 16 bits or four hexadecimals

❖ Can be written in either lowercase or uppercase

❖Rule 1- Omitting Leading 0s

- ➢The first rule to help reduce the notation of IPv6 addresses is any leading 0s (zeros) in any 16-bit section or hextet can be omitted
  - 01AB can be represented as 1AB
  - 09F0 can be represented as 9F0
  - 0A00 can be represented as A00

Preferred	2001:0DB8:0000:1111:0000:0000:0000:0200
No leading 0s	2001: DB8: 0:1111: 0: 0: 0: 200

Preferred	0000:0000:0000:0000:0000:0000:0000:0001
No leading 0s	0: 0: 0: 0: 0: 0: 0: 1

❖Rule 2- Omitting All 0 Segments

- ➢A **double colon** (::) can replace any single, contiguous string of one or more 16-bit segments (hextets) consisting of all 0's
- ➢Known as the *compressed format*
- ➢Double colon (::) can only be used **once** within an address otherwise the address will be ambiguous.

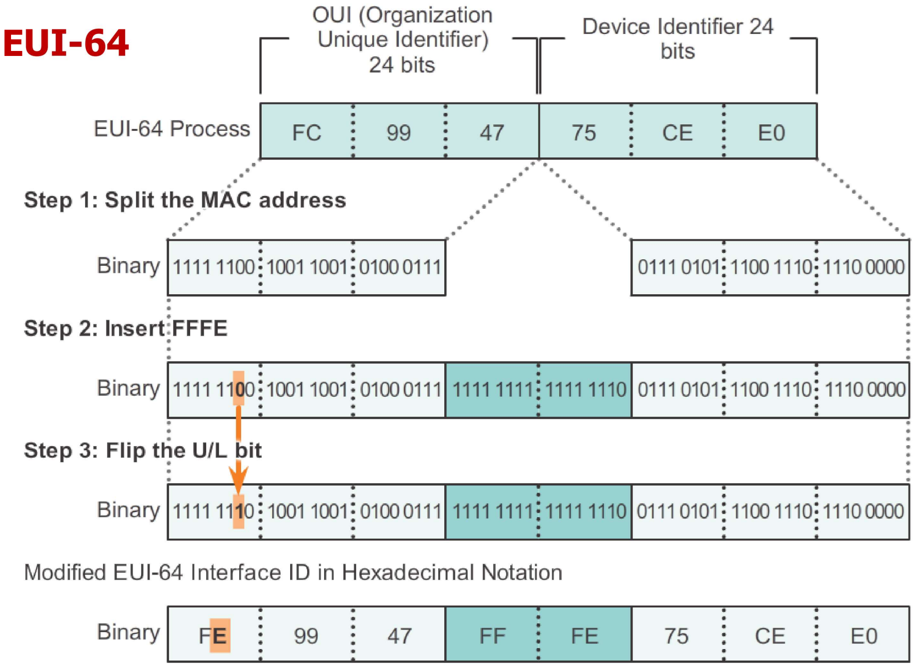
Example: 1843:f01::22::fa

1843:0f01:0000:0000:0022:0000:0000:00fa ?
1843:0f01:0000:0000:0000:0022:0000:00fa ?
1843:0f01:0000:0022:0000:0000:0000:00fa ?

❖Interface identifiers

- ➢IPv6 addresses can be divided into a 64-bit network prefix and a 64-bit "host" portion.
  - o host-portion bits are known officially as the interface identifier.
- ➢Original plan for the interface identifier was to derive it in most cases from the LAN address.
  - o 48 bits → 64 bits ???
    - **EUI-64**

❖EUI-64



2. IPv6 Addresses

❖**EUI-64**

➢Review MAC Address:

```
0        8        16       24       32       40       48 (bit)
+--------+--------+--------+--------+--------+--------+
|ccccccUG:cccccccc:cccccccc:mmmmmmmm:mmmmmmmm:mmmmmmmm|
+--------+--------+--------+--------+--------+--------+
```

- **"c"** is the bits of the assigned company_id,
- **"m"** is the bits of the manufacturer selected extension identifier.
- **"U"** is the value of the Universal/Local bit.
  - 0: Universal, i.e., MAC address is the burned-in-address (BIA)
  - 1: Local, i.e., MAC address that has been changed locally
- **"G"** is individual/group bit
  - 0: individual (i.e., unicast)
  - 1: group (i.e., multicast/broadcast)

2. IPv6 Addresses

❖**EUI-64**

➢A **major privacy concern**:

- No matter where a host connects to the Internet – home or work or airport or Internet cafe
  - Such an interface identifier always remains the same, and thus serves as a permanent host fingerprint.

➢Now discouraged for personal workstations and mobile devices. (Only for some fixed-location hosts)

2. IPv6 Addresses

❖**An alternative (RFC 7217):**

➢The interface identifier is a **secure hash** of:

- A "Net_Iface" parameter.
  - The interface's MAC address,
  - Or the interface's "name", e.g., eth0.
- 64-bit IPv6 address prefix.
- A host-specific secret key.

➢Interface identifiers created this way change from connection point to connection point (because the prefix changes).

2. IPv6 Addresses

❖**Link-local addresses:**

➢Used only on a single LAN and **never routed**.

➢ **FE80::/64**

- The remaining 64 bits are the interface identifier (EUI-64 or "Hashing" methods).

➢**Applications:**

- A "bootstrap" address for global-address auto-configuration
- An optional permanent address for routers.
  - IPv6 routers often communicate with neighboring routers via their link-local addresses.

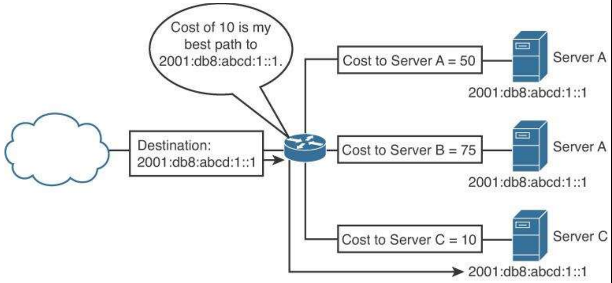
2. IPv6 Addresses

❖**Anycast addresses:**

➢Assigned to each of a set of routers/servers.

➢A packet addressed to this anycast address would be delivered to only one member of this set.

➢It is up to the local routing infrastructure to decide which member of the anycast group would receive the packet; normally it would be sent to the "closest" member.



https://www.ciscopress.com/articles/article.asp?p=2803866&seqNum=6

3 Network Prefixes

3. Network Prefixes

❖IPv6 address is composed of a 64-bit network prefix and a 64-bit interface identifier.

❖Routers still uses **CIDR** (10.1 Classless Internet Domain Routing: CIDR) and still bases forwarding decisions on prefixes shorter than /64.

❖64-bit network prefixes are supplied by a provider, and which represent the first half of globally routable IPv6 addresses.

➢These are the prefixes that will be visible to the outside world.

3. Network Prefixes

❖IPv6 customers will typically be assigned a relatively large block of addresses, *e.g.,* "/48" or "/56".

➢The former allows 64–48 = 16 bits for local "subnet" specification within a 64-bit network prefix

➢The latter ("/56") allows 8 subnet bits.

2. IPv6 Addresses

❖**Unique Local** Unicast Address:

- ➢Corresponding to IPv4 private address blocks like 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8.
- ➢The first 8 bits of a unique-local prefix are 1111 1101 (fd00::/8).
- ➢The related prefix 1111 1100 (fc00::/8) is reserved for future use.
- ➢The last 16 bits of a 64-bit unique-local prefix represent the subnet ID, and are assigned either administratively or via autoconfiguration.
- ➢The 40 bits in between, from bit 8 up to bit 48, represent the Global ID.
  - ○A site is to set the Global ID to a pseudorandom value.

4 IPv6 Multicast

4. IPv6 Multicast

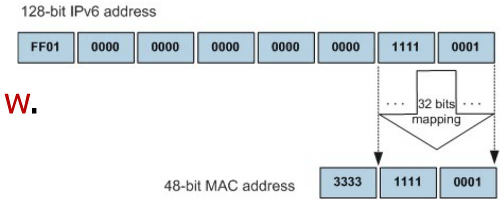
❖IPv6 has no LAN-layer broadcast.

- ➢Providing a wide range of LAN-layer multicast groups.
- ➢Intended to limit broadcast traffic in general, though many switches still propagate LAN multicast traffic everywhere, like broadcast.
- ➢**ff00::/8**

4. IPv6 Multicast

❖For actual delivery, IPv6 multicast addresses correspond to LAN-layer (e.g., Ethernet) multicast addresses through a well-defined static correspondence.

- ➢If x, y, z and w are the last four bytes of the IPv6 multicast address, in hex,
  - →Ethernet multicast address is **33:33**:x:y:z:w.



❖**ff02::1**

- ➢All-nodes
- ➢Ethernet multicast address: **33:33**:00:00:00:01.

❖**ff02::2**

- ➢All-routers.
- ➢Ethernet multicast address: **33:33**:00:00:00:02.

4. IPv6 Multicast

- ❖IPv6 nodes on Ethernets send LAN-layer Multicast Listener Discovery (MLD) messages to multicast groups they wish to start using.
  - ➢These messages allow multicast-aware Ethernet switches to optimize forwarding so that only those hosts that have subscribed to the multicast group in question will receive the messages.
  - ➢Otherwise, switches are supposed to treat multicast like broadcast;
  - ➢Worse case, some switches may simply fail to forward multicast packets to destinations that have not explicitly opted to join the group.

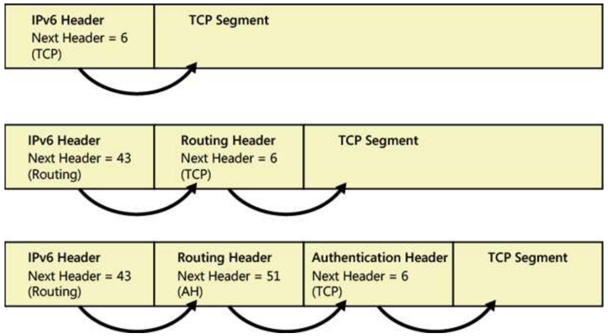
5 IPv6 Extension Headers

5. IPv6 Extension Headers

- ❖Hop-by-Hop options header
- ❖Destination options header
- ❖Routing header
- ❖Fragment header
- ❖Authentication header
- ❖Mobility header
- ❖Encapsulated Security Payload header

5. IPv6 Extension Headers

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Header  |  Hdr Ext Len  |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                                                               |
.                                                               .
.                  Header Specific Data                         .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```



❖Hop-by-Hop options header

- Consists of a set of ***<type,value>*** pairs which are intended to be processed by each router on the path.
  - The only Hop-by-Hop options provided by RFC 2460 were for padding, so as to set the alignment of later headers.
  - RFC 2675 later defined a Hop-by-Hop option to support IPv6 **jumbograms**: datagrams larger than 65,535 bytes.
- Because this options headers must be processed by each router encountered, they have the potential to overburden the Internet routing system. As a result, RFC 6564 strongly discourages new Hop-by-Hop Option headers, unless examination at every hop is essential.

❖Destination Options Header

- Consists of a set of ***<type,value>*** pairs, and the original RFC 2460 specification only defined options for padding. (similar to the Hop-by-Hop Options header)
- Since **RFC 2460**, a few more Destination Options header types have been defined, though none is in common use.
- **RFC 2473** defined a Destination Options header to limit the nesting of tunnels, called the Tunnel Encapsulation Limit.
- **RFC 6275** defines a Destination Options header for use in Mobile IPv6.
- **RFC 6553**, on the Routing Protocol for Low-Power and Lossy Networks (RPL), has defined a Destination (and Hop-by-Hop) Options type for carrying RPL data.

❖Routing Header

- The original, or **Type 0**, Routing header contained a list of IPv6 addresses through which the packet should be routed. These did not have to be contiguous.
  - Ex: The list to be visited enroute to destination ***D*** was $<R_1, R_2, \ldots, R_n>$.
    - Then this option header contained $<R_2, R_3, \ldots, R_n, D>$ with ***R1*** as the initial destination address;
    - ***R1*** then would update this header to $<R_1, R_3, \ldots, R_n, D>$, and would send the packet on to ***R2***.
    - This was to continue on until R_n addressed the packet to the final destination ***D***.

❖Routing Header

- The original, or **Type 0**, Routing header contained a list of IPv6 addresses through which the packet should be routed. These did not have to be contiguous.
  - Ex: (continue)
    - The header contained a Segments Left pointer indicating the next address to be processed, incremented at each R_i. When the packet arrived at ***D*** the ***Routing Header*** would contain the routing list $<R_1, R_2, \ldots, R_n>$.
    - Note, however, that routers between the listed routers $R_1 \ldots R_n$ did not need to examine this header; they processed the packet based only on its current destination address.

5. IPv6 Extension Headers

❖**Routing Header**

- This form of routing header was deprecated by **RFC 5095**, due to concerns about a traffic-amplification attack.
  - An attacker could send off a packet with a routing header containing an alternating list of just two routers ***<R$_1$,R$_2$,R$_1$,R$_2$,...,R$_1$,R$_2$,D>.***
    - This would generate substantial traffic on the R1–R2 link.

5. IPv6 Extension Headers

❖**Routing Header**

- **RFC 6275** and **RFC 6554** define more limited routing headers.
  - **RFC 6275** defines a quite limited routing header to be used for IPv6 mobility (and also defines the IPv6 Mobility header).
  - **RFC 6554** routing header used for RPL, mentioned above, has the same basic form as the Type 0 header described above, but its use is limited to specific low-power routing domains.

5. IPv6 Extension Headers

❖**IPv6 Fragment Header**

- Contains a 13-bit Fragment Offset field (as in IPv4) and a 32-bit Identification field.
- IPv6 fragmentation is done only by the original sender.
- Generally speaking, fragmentation should be avoided at the application layer when possible.

6 Neighbor Discovery

6. Neighbor Discovery

- A set of related protocols that replaces several IPv4 tools, most notably ARP, ICMP redirects and most non-address-assignment parts of DHCP.

- Provides the following services:
  - Finding the local router(s)
  - Finding the set of network address prefixes that can be reached via local delivery.
  - Finding a local host's LAN address, given its IPv6 address.
  - Detecting duplicate IPv6 addresses.
  - Determining that some neighbors are now unreachable

6. Neighbor Discovery

- **Router Discovery**
  - IPv6 routers periodically send Router Advertisement (RA) packets to the all-nodes multicast group.
    - Hosts can request an RA packet immediately by sending a Router Solicitation (RS) request to the all-routers multicast group.
  - RA packets contain:
    - Identification of the routers,
    - A list of all network address prefixes in use on the LAN.
    - An agreed-on MTU.

6. Neighbor Discovery

- **Prefix Discovery**
  - Support hosts to learn the network-address prefixes.
  - Router Advertisement packets contain a complete list of valid network-address prefixes, as the Prefix Information option.
  - Each prefix will have an associated lifetime.
    - Nodes receiving a prefix from an RA packet are to use it only for the duration of this lifetime.
    - On expiration a node must obtain a newer RA packet with a newer prefix list.

6. Neighbor Discovery

- **Neighbor Solicitation** (Similar to IPv4 ARP requests)
  - Sent to the solicited-node multicast address.
    - ff02::0001:x.y.z.w.
      - x, y, z and w are the low-order 32 bits of the IPv6 address the sender is trying to look up.
    - Each IPv6 host on the LAN will need to subscribe to all the solicited-node multicast addresses corresponding to its own IPv6 addresses.
  - Neighbor Solicitation messages are repeated regularly but follow up verifications are initially sent to the unicast LAN address on file.
  - The target host's response to a Neighbor Solicitation message is called **Neighbor Advertisement**.

7 IPv6 Host Address Assignment

❖IPv6 provides two competing ways for hosts to obtain their full IP addresses.

➢DHCPv6.

➢StateLess Address Auto Configuration (SLAAC)

❖Duplicate Address Detection

➢Whenever an IPv6 host obtains a unicast address it goes through a duplicate-address detection (DAD) process.

➢The host sends one or more Neighbor Solicitation messages, asking if any other host has this address.

- If anyone answers, then the address is a duplicate.

❖Duplicate Address Detection

➢A host is not allowed to use an IPv6 address if the DAD process has failed.

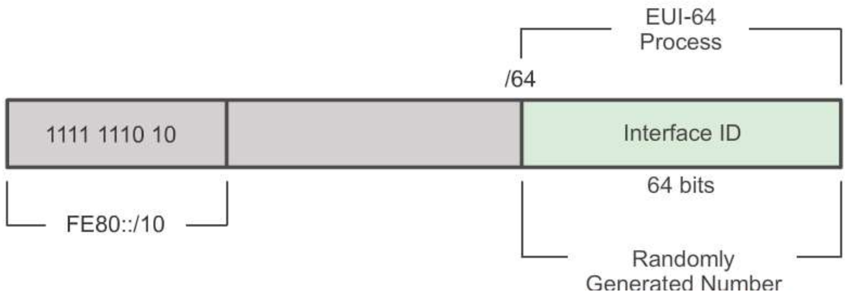
- If the DAD process fails for an address based on an EUI-64 identifier, then some other node has the same Ethernet address and you have bigger problems than just finding a working IPv6 address.

- If the DAD process fails for an address constructed with the Interface identifiers, the host is able to generate a new interface identifier and try again.

7. IPv6 Host Address Assignment

❖Stateless Autoconfiguration (SLAAC)

- ➢Host generates its link-local address, appending the standard 64-bit link-local prefix fe80::/64 to its interface identifier.
- ➢Derived from the host's LAN address using either EUI-64 or the RFC 7217 mechanism.



- ➢Do DAD.

7. IPv6 Host Address Assignment

❖Stateless Autoconfiguration (SLAAC)

- ➢If there is a router available.
  - The **host** send a RS message.
  - A **router** answers with a RA message that contains a Prefix Information option.
    - RA message will mark with a flag those prefixes eligible for use with SLAAC.
    - All prefixes will also be marked with a lifetime.
  - The **host** chooses an appropriate prefix, stores the prefix-lifetime information, and appends the prefix to the front of its interface identifier
    - This is a routable address.
  - Do DAD.

7. IPv6 Host Address Assignment

❖Stateless Autoconfiguration (SLAAC)

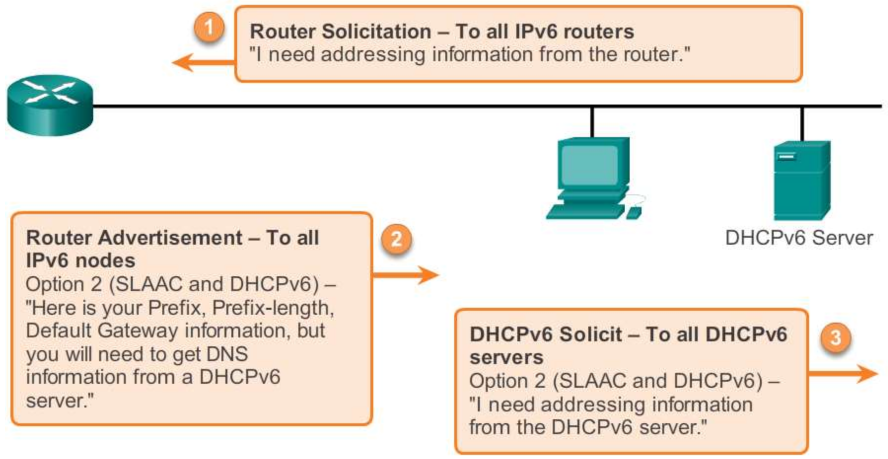
- ➢The **host** knows its IPv6 address and its default router but no DNS server.
  - In **RFC 6106** now defines a process by which IPv6 routers can include DNS-server information in the RA packets they send to hosts as part of the SLAAC process.

7. IPv6 Host Address Assignment

❖DHCPv6

- ➢Similar to IPv4, host automatically receives addressing information including a global unicast address, prefix length, default gateway address and DNS server addresses
- ➢Device receives all or some of its IPv6 addressing information from a DHCPv6 server depending upon option 2 (SLAAC and DHCPv6) or option 3 (DHCPv6 only) is specified in the ICMPv6 RA message.
- ➢A device may obtain its IPv6 global unicast address dynamically and also be configured with multiple static IPv6 addresses on the same interface.

❖DHCPv6



Note: An RA with option 3 (DHCPv6 Only) enabled will require the client to obtain all information from the DHCPv6 Server.

8 ICMPv6

8. ICMPv6

- ❖As part of the Neighbor Discovery Protocol.
- ❖Messages are identified by 8-bit type and code (subtype) fields.
- ❖Distinguishes between informational and error messages by the first bit of the type field.
- ❖ICMPv6 includes an IPv6 version of Echo Request/Echo Reply, upon which the “ping6” command.

❖**Destination Unreachable**

➢In this case, one of the following numeric codes is returned:

0: No route to destination, returned when a router has no *next_hop* entry.

1: Communication with destination administratively prohibited, returned when a router has a *next_hop* entry, but declines to use it for policy reasons. Codes 5 and 6, below, are special cases of this situation; these more-specific codes are returned when appropriate.

2: Beyond scope of source address, returned when a router is, for example, asked to route a packet to a global address, but the return address is not, e.g., is unique-local. In IPv4, when a host with a private address attempts to connect to a global address, NAT is almost always involved.

❖**Destination Unreachable**

➢In this case, one of the following numeric codes is returned:

3: Address unreachable, a catchall category for routing failure not covered by any other message. An example is if the packet was successfully routed to the last_hop router, but Neighbor Discovery failed to find a LAN address corresponding to the IPv6 address.

4: Port unreachable, returned when, as in ICMPv4, the destination host does not have the requested UDP port open.

5: Source address failed ingress/egress policy, see code 1

6: Reject route to destination, see code 1.

❖**Packet Too Big**

➢This is like ICMPv4's "Fragmentation Required but Don't Fragment flag set"; IPv6 however has no router-based fragmentation.

❖**Time Exceeded**

➢This is used for cases where the Hop Limit was exceeded, and also where *source*-based fragmentation was used and the fragment-reassembly timer expired.

❖**Parameter Problem**

➢This is used when there is a malformed entry in the IPv6 header, an unrecognized Next Header type, or an unrecognized IPv6 option.

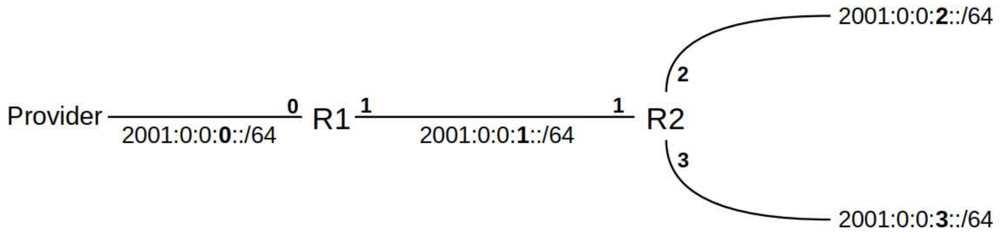
9 IPv6 Subnets

9. IPv6 Subnets

❖In IPv6, this is much simpler: all subnets are of size /64.

➢There is one common exception: RFC 6164 permits the use of 127-bit prefixes at each end of a point-to-point link. The 128th bit is then 0 at one end and 1 at the other.

❖A site receiving from its provider an address prefix of size /56 can assign up to 256 "/64 subnets".



9. IPv6 Subnets

❖If users *are* given only "/64" blocks, and they want to use subnets.

➢They have to break the 64/64 rule locally.

- Perhaps they can create four subnets each with a prefix of length 66 bits, and each with only 62 bits for the host identifier.
- Wanting to do that in a standard way would dictate more flexibility in the prefix/host division.

QA

Lecturer: Nguyen Viet Ha, Ph.D.
Ho Chi Minh City University of Science
Faculty of Electronics and Communications
Department of Telecommunication and Networks
Email: nvha@fetel.hcmus.edu.vn