IPv6 Specification

簡介：Some changes from IPv4 to IPv6:

- **Expanded Addressing Capabilities** (RFC 1884)
  - from 32 bits to 128 bits (more level and nodes)
  - improve multicast routing ("scope" field)
  - "anycast address": send a packet to any one of a group of nodes
- **Header Format Simplification**
  - limit bandwidth cost
- **Extensions and Options**
  - more flexibility
- **Flow Labeling Capability**
  - define traffic "flow" to support special QoS
- **Authentication and Privacy Capabilities**
  - authentication, data integrity, and data confidentiality

IPv6 Header Format:

32 bits

<table>
<tr><td>Version</td><td>Prio.</td><td colspan="3">Flow Label</td></tr>
<tr><td colspan="3">Payload Length</td><td>Next Header</td><td>Hop Limit</td></tr>
<tr><td colspan="5">Source address</td></tr>
<tr><td colspan="5">Destination address</td></tr>
</table>

IPv6 Extension Headers:

Example:

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 = Fragment	Fragment header Next Header = TCP	Fragment of TCP Header + data

⇨Hop-by-Hop Options, Routing, Fragment, Destination Options, Authentication (RFC 1826), Encapsulating Security Payload (RFC 1827)

⇨除了 hop-by-hop options 外, 其他的 extension headers 不可以被在傳送路徑中的其他非 destination 的 nodes 檢查與處理

⇨each extension header 的內容與語意決定是否要處理下一個 header，因此，extension header must processed strictly in Order

⇨Extension Header Order:

IPv6
Hop-By-Hop Options header (0)
Destination Options header (60)
Routing header (43)
Fragment header (44)
Authentication header
Encapsulating Security Payload header
Destination Options header
Upper-layer header

Note: TCP (6), UDP(17), Nothing(59)

⇨錯誤處理：ICMP (RFC 1885)

⇨Each extension header occurs at most once, except for Destination Options header which occurs at most twice

Options:

⇨Hop-by-Hop Options header and Destination Options header carry a variable number type-length-value (TLV) encoded "options"

⇨Option format:

Option Type	Opt Data Len	Option Data

⇨Option type:

highest-order two bits **表示當無法辨識此** option type **時所要採取的行動**

00 —skip this option and continue processing the header

01 —discard the packet

10 —discard the packet and send ICMP message to source

11 —discard the packet and if Destination address was not a multicast address send ICMP message to source

third-highest-order bit **表示在傳送的過程中** Option data **是否被改變**。

0 —Option data doesn' t change en-route

1 —Option data may change en-route

Alignment:

Notation xn+y : option type must appear at an integer multiple of x bytes from the start of the header, plus y bytes.

Padding options:

Pad1 option : used to insert one byte of padding into the option area

One byte, special option format: no length and value field

PadN option : used to insert more than one byte

1	Opt Len	Opt data = 0 …

Example: 4n+3 (hop-by-hop options)

Next Header	Hdr Ext Len =1	Pad1 Option=0	Option type=Y
Opt Data Len=7	1-Byte field	2-byte filed	
4-byte field			
Pad4 Option=1	Opt Len=2	0	0

Hop-by-Hop Options header:

Next header	Hdr Ext len	
Options (multiple of 8 bytes)		

Jumbo Payload option (4n+2) – hop-by-hop option

	194	Opt Data Len=4
Jumbo Payload length		

⇨Used to send packets with payload longer than 65535 bytes.

⇨Payload length in IPv6 header must be zero

⇨Jumbo payload options must not be used in a packet that carries a Fragment header

⇨ Links' MTU must be greater than 65575 = 65535+40

⇨**功用: 如** the special handling by routers (source desires)

Routing Header

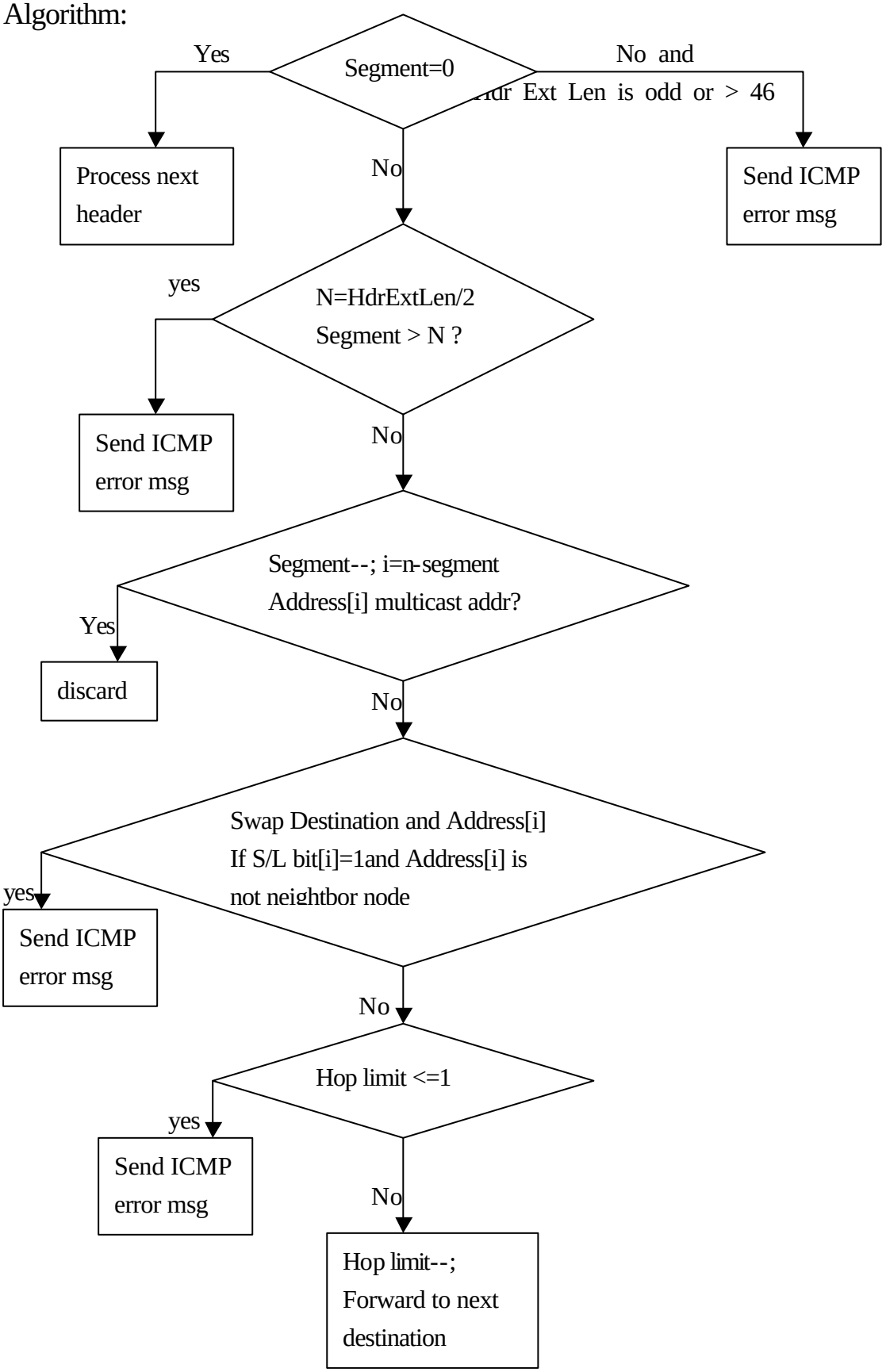
Next header	Hdr Ext Len	Routing Type	Segment Left
Type-specific data			

Type 0 Routing header:

Next header	Hdr Ext Len	Routing Type=0	Segment Left
Reserved	Strict/Loose Bit Map		
Address[1]			
…			
Address[n]			

⇨Segment Left: Maximum legal number = 23
⇨24-bits bit-map, number 0 to 23, left-to-right, indicate whether or not the next destination address must be a neighbor: 1–strict, 0–loose

Algorithm:


Segment=0
Yes
No and
Hdr Ext Len is odd or > 46
Process next header
Send ICMP error msg
No
yes
N=HdrExtLen/2
Segment > N ?
Send ICMP error msg
No
Segment--; i=n-segment
Address[i] multicast addr?
Yes
discard
No
Swap Destination and Address[i]
If S/L bit[i]=1and Address[i] is not neighbor node
yes
Send ICMP error msg
No
Hop limit <=1
yes
Send ICMP error msg
No
Hop limit--;
Forward to next destination

Example:
Source: S, Destination: D Intermediates: I1, I2, I3

As the packet travels from S to I1

Source address = S	Hdr Ext Len = 6
Destination=I1	Segment left = 3
(if bit 0 of Bit Map is 1,	Address[1]=I2
S and I1 must be neighbors;	Address[2]=I3
This is checked by S)	Address[3]=D

As the packet travels from I1 to I2

Source address = S	Hdr Ext Len = 6
Destination = I2	Segment left = 2
(if bit 1…)	Address[1]=I1
	Address[2]=I3
	Address[3]=D

As the packet travels from I2 to I3

Source address = S	Hdr Ext Len = 6
Destination = I3	Segment left = 1
(if bit 2 ….).	Address[1]=I1
	Address[2]=I2
	Address[3]=D

As the packet travels from I3 to D

Source address = S	Hdr Ext Len = 6
Destination = D	Segment left = 0
(if bit 3 ..).	Address[1]=I1
	Address[2]=I2
	Address[3]=I3

Fragment Header:

⇨ fragmentation in IPv6 is performance only by Source

Next Header	Reserved	Fragment offset	Res	M
Identification				

⇨Every packet must have different identification, increase each time when a packet must be fragmented

original packet:

Unfragmentable Part	Fragmentable Part

⇨Unfragmentable Part: IPv6 header and extension headers that must be processed by nodes en-route to the destination

⇨Fragmentable Part: extension header that need be processed only by the final destination, plus upper–layer header and data

original packet:

Unfragmentable part	First fragment	Second fragment	…..		Last fragment

Fragment packets:

Unfragmentable part	Fragment Header	First fragment

.

.

Unfragmentable part	Fragment Header	Last fragment

Each packet is composed of:

(1) Unfragmentable part:

payload length = fragment packet only

(2) Fragment header:

Fragment offset, M flag, Identification

(3) Fragment

⇨Reassemble packet from fragment, packets must have the same Source Address, Destination Address, and Fragment Identification

⇨Error conditions:

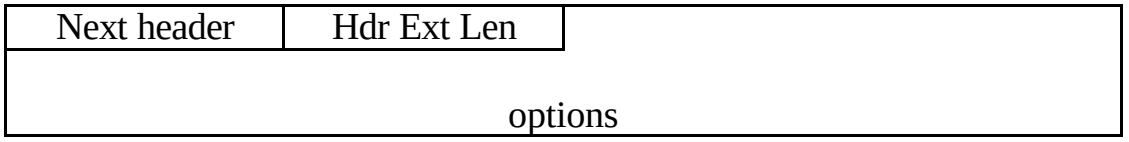
(1)insufficient fragments are received to complete reassembly of a packet within 60 seconds

(2)flag M = 1 and payload length is not a multiple of 8 bytes

(3)payload length of fragment > 65535

Destination Options Header

⇨used to carry optional information that need be examined only by destination



⇨two way to encode optional destination information: Destination Options Header, or as a separate extension header (fragment/authentication)

Packet size issues:

⇨MTU of every link must >= 576 bytes, **如果無法提供，則必須靠下層來做** fragmentation and reassembly

⇨**可以利用** Path MTU Discovery[RFC-1191] to discover and take advantage of path with MTU greater than 576 bytes

⇨**最簡單的方法就是限制** packet **的** size <=576 bytes

⇨a node must not send fragments that reassemble to a size greater than 1,500 byte.

⇨IPv6 packet is sent to an IPv4 destination.

Source may receive ICMP message.

Subsequent packets to 528 (payload=576-40-8) and include Fragment header, so that the IPv6-to-IPv4 translating router can know.

Flow Labels

⇨A flow: a sequence of packets which the source desires special handling (by hop-by-hop option or control protocol)

⇨identified by combination of source address and non-zero flow label(randomly and uniformly chosen), as a hash key used by router.

⇨All packets belonging to the same flow must have the same source address, destination address, priority, and flow label

⇨The router may cache the information of next-hop interface, how to queue the packet based on its priority, etc…

Priority:

0–7: provide congestion control
8–15: no back off in response to congestion, real-time packets

8 : most willing to discard under congestion (e.g. high-fidelity video traffic)
15: least willing …(e.g. low-fidelity audio traffic)

Upper-layer Protocol Issues

⇨Upper-layer Checksums – Pseudo-header

⇨Maximum Packet Lifetime – don' t enforce maximum packet lifetime

⇨Upper-layer payload length –
IPv4 : max packet size – 40 (20 for IP header, 20 for TCP header)
IPv6 : max packet size – 60