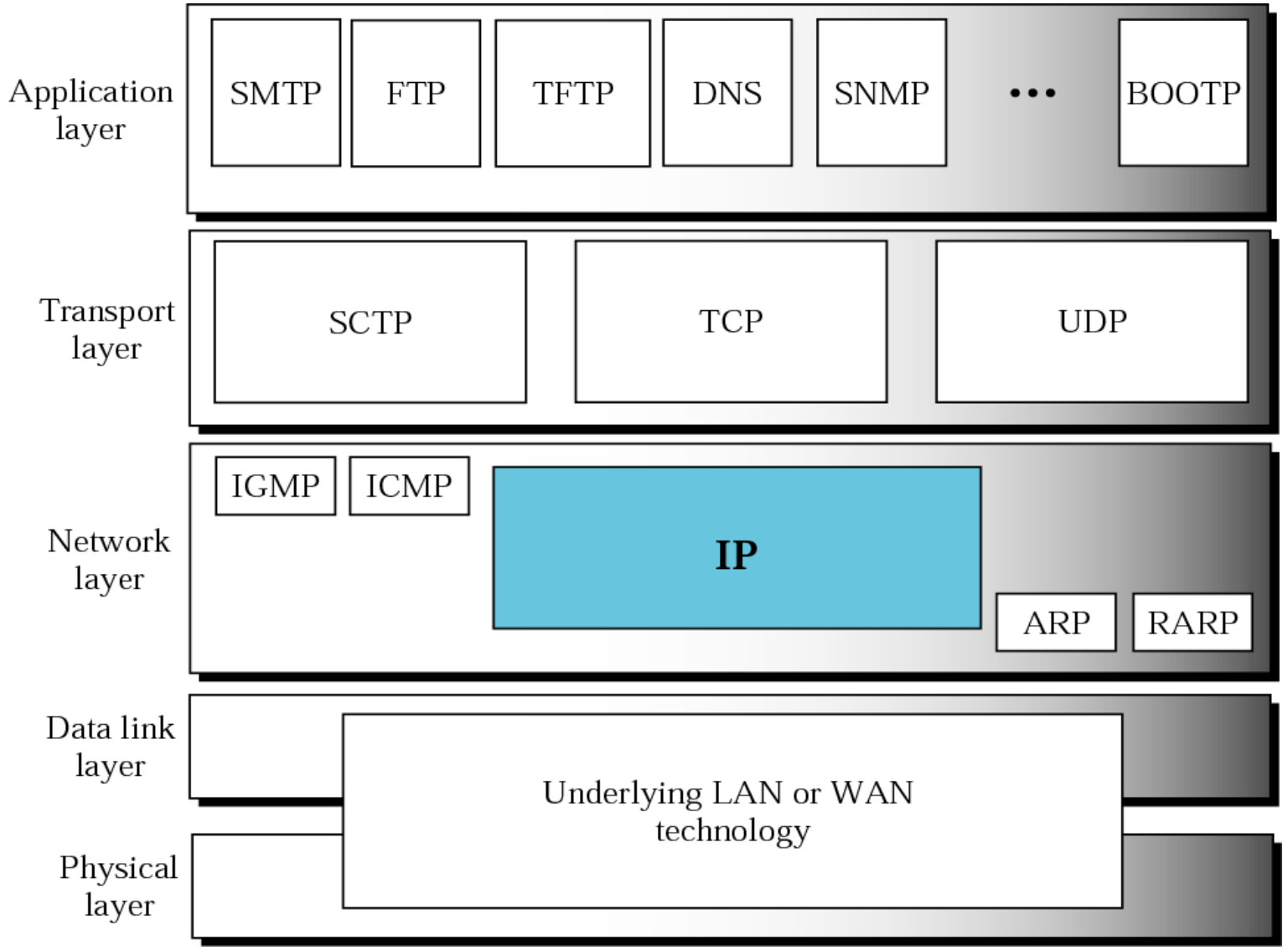
Chapter -3
Internet Protocols

Points to be covered

- IPv4 Protocol
- ICMPv4
- IPv6 Protocol
- ICMPv6
- Transition from IPv4 to IPv6

TCP/IP protocol suit



About IP

- IP by itself can be compared to something like the postal system.

- It allows you to address a package and drop it in the system, but there's no direct link between you and the recipient.

- TCP/IP, on the other hand, establishes a connection between two hosts so that they can send messages back and forth for a period of time.

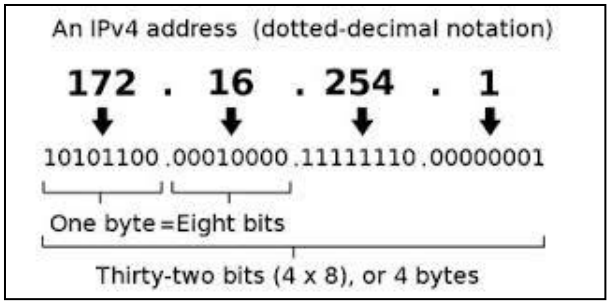
Internet Protocol Version 4

IPv4

About IP*v*4

- IPv4 stands for Internet Protocol version 4.

- It is the underlying technology that makes it possible for us to connect our devices to the web. Whenever a device access the Internet (whether it's a PC, Mac, smartphone or other device), it is assigned a unique, numerical IP address such as 99.48.227.227.



- To send data from one computer to another through the web, a data packet must be transferred across the network containing the IP addresses of both devices.

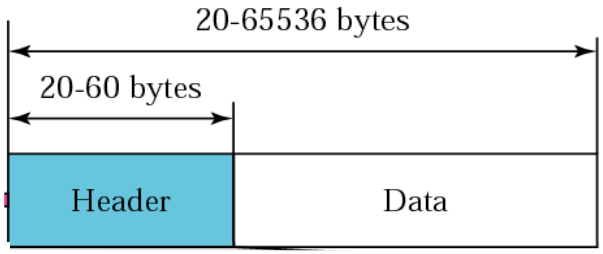
About IP*v*4

- A *packet* in the IP layer is called a *datagram*.
- Datagram travel along with *different routes* & may arrive at *different time.*
- IP is Unreliable, connectionless *datagram* protocol.
  - Since IP packets can be corrupted, lost, arrive out of order or delayed and may create congestion.
  - For e.g. Post Office
- For reliability – *pair it with TCP*
- Minimum *size* of IP datagram is of *20bytes.*
- Maximum *size* of IP datagram is of *65535bytes.*
- IP relies on higher level protocol to take care of transmission related issues.

IPv4 datagram Format

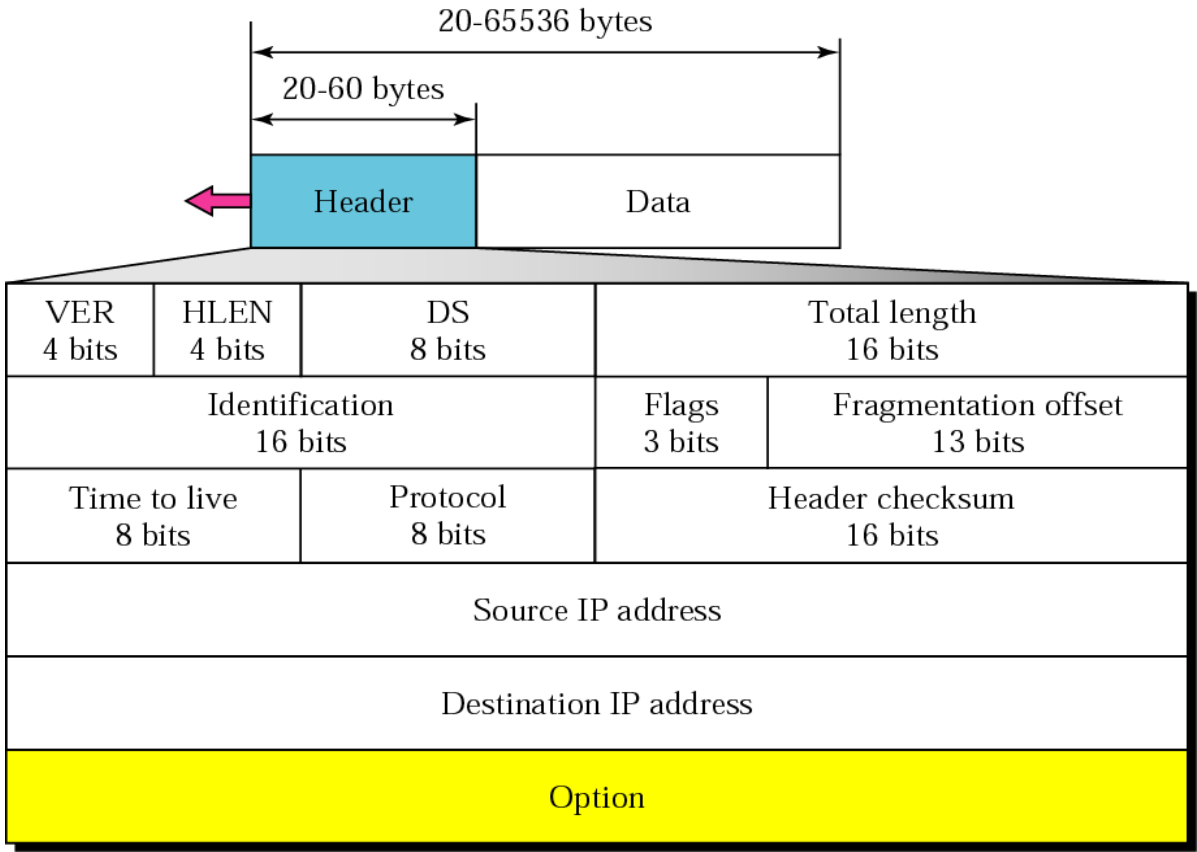
IP datagram total size :::: **20 to 65536bytes**

- Header – 20-60 bytes
- Data - remaining



IPv4 datagram Format

- Version
- Header length
- Differentiated service
- Total length
- Identification
- Flags
- Fragmentation offset
- Time to live
- Protocol
- Checksum
- Source address
- Destination address



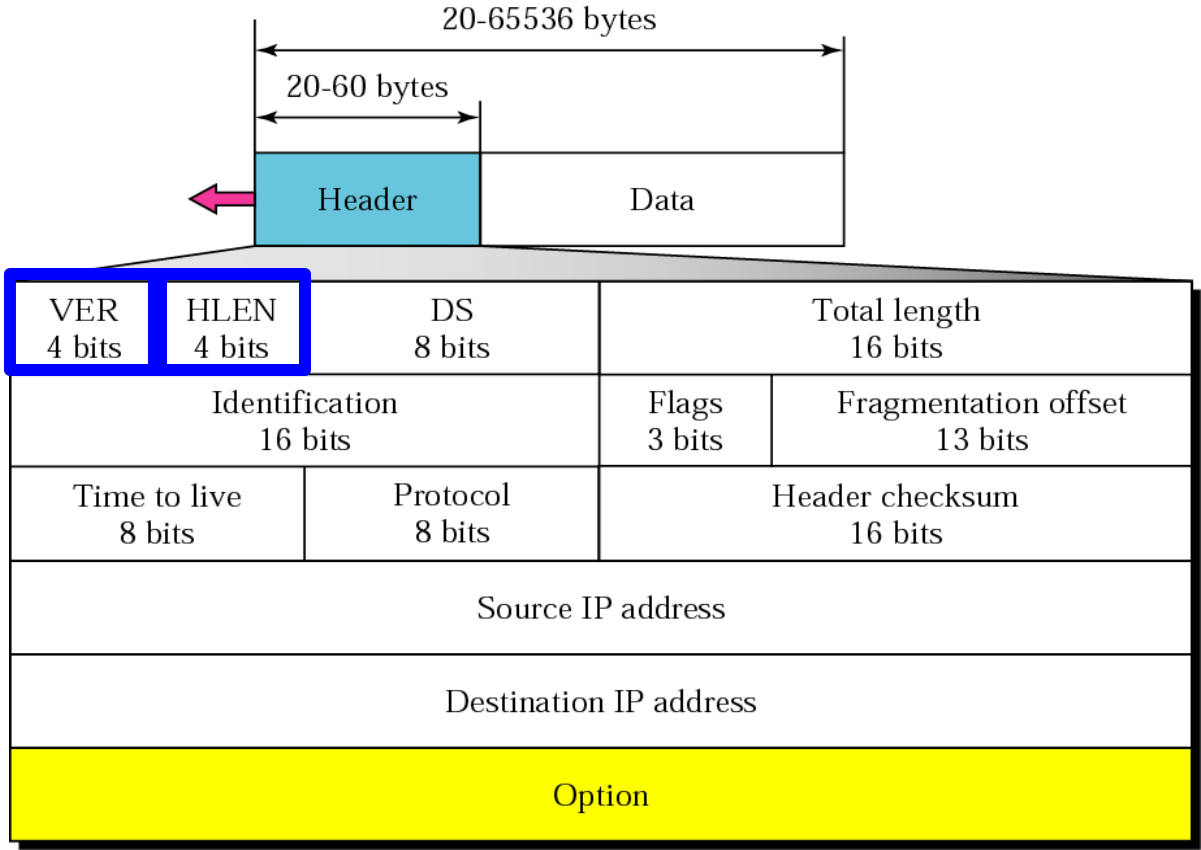
IPv4 datagram Format

Version

- 4 bit field defines **version of IP protocol** used.

Header length (HLEN)

- 4 bit field defines total **length of datagram header** since header length is variable



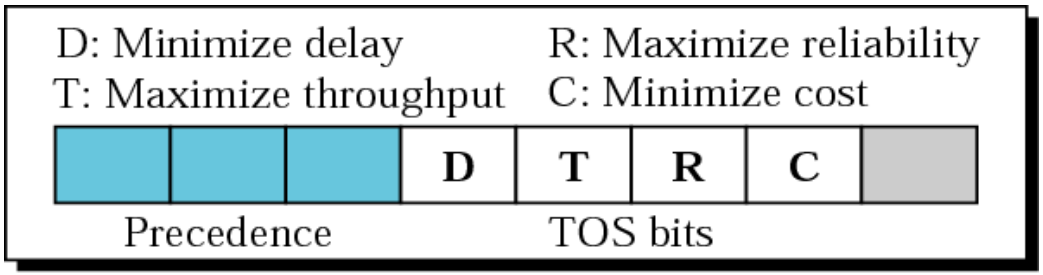
IPv4 datagram Format

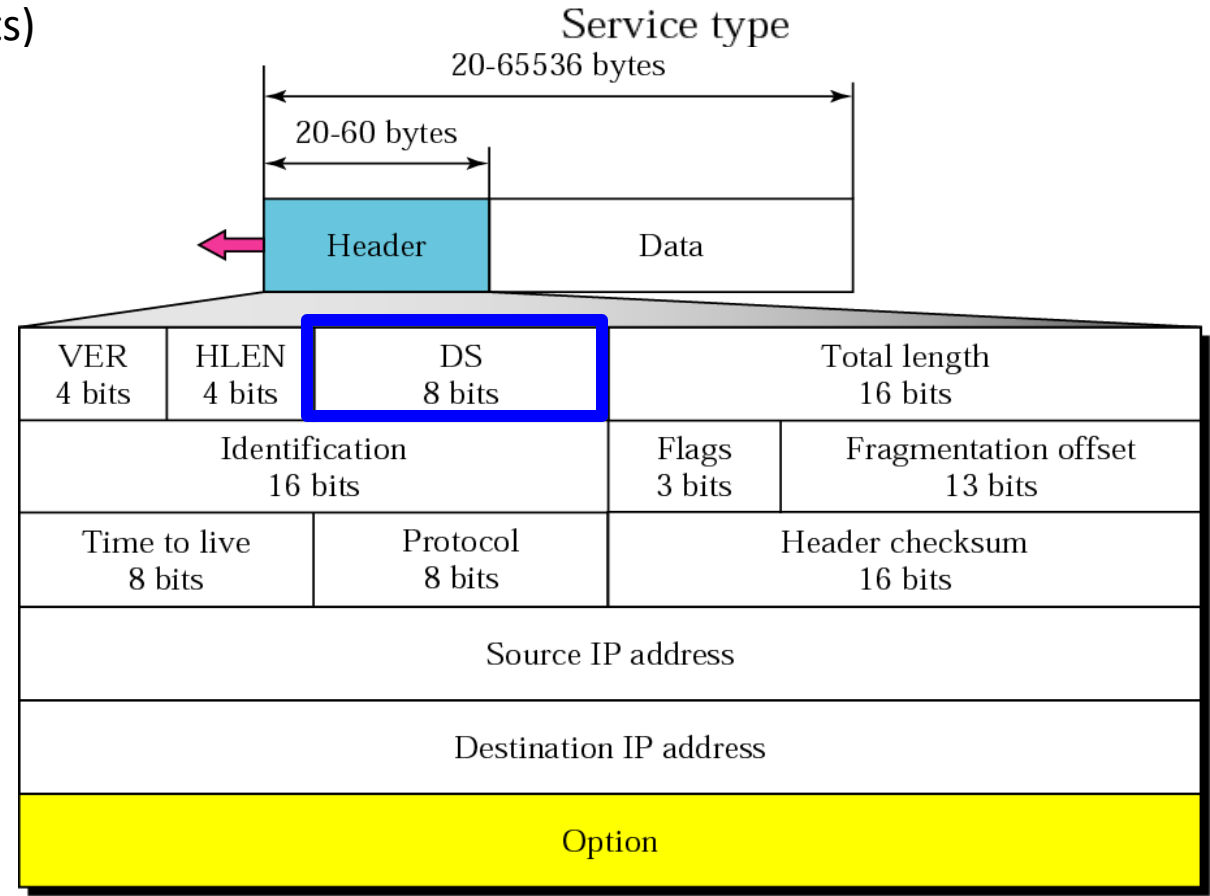
Differentiated service: (8 bit field)

Service type :

Precedence : (3 bits)defines priority.

TOS bits (type of service) : (4 bits)





Default types of service

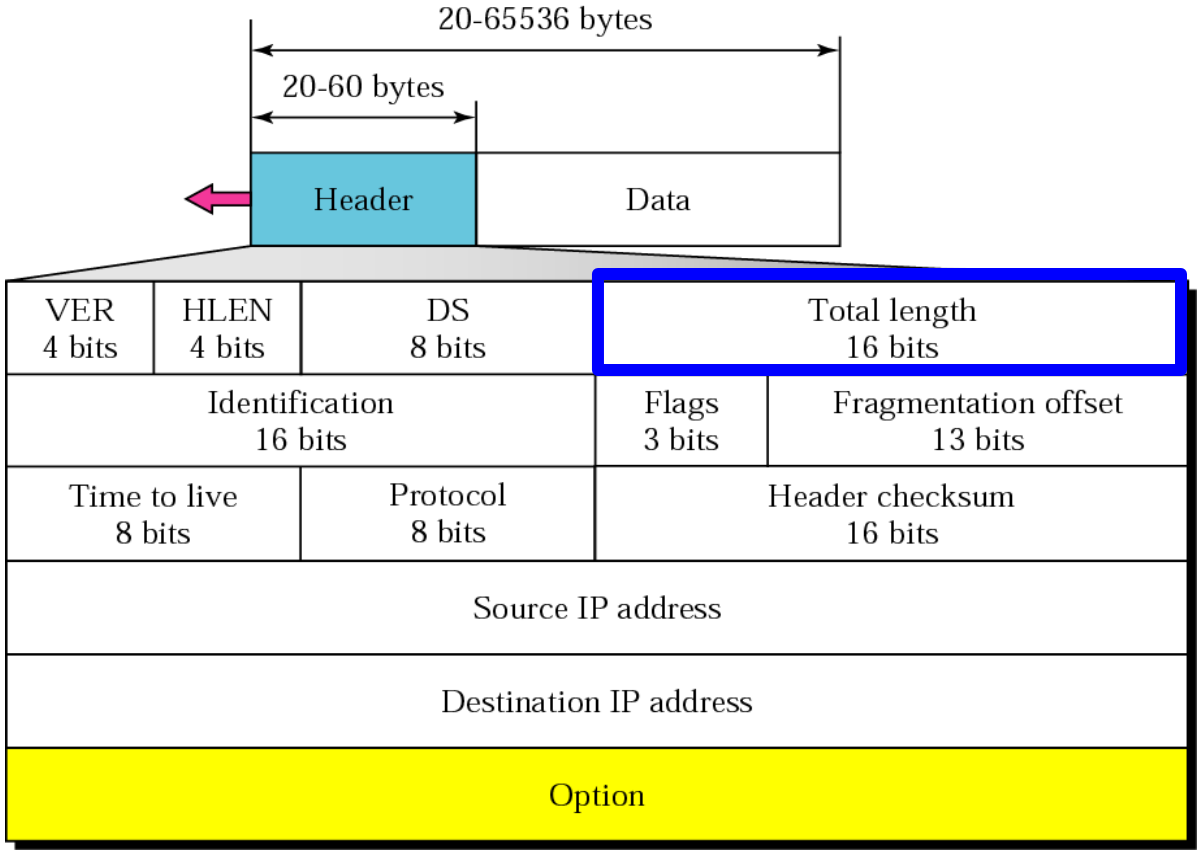
TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

IPv4 datagram Format

Total Length

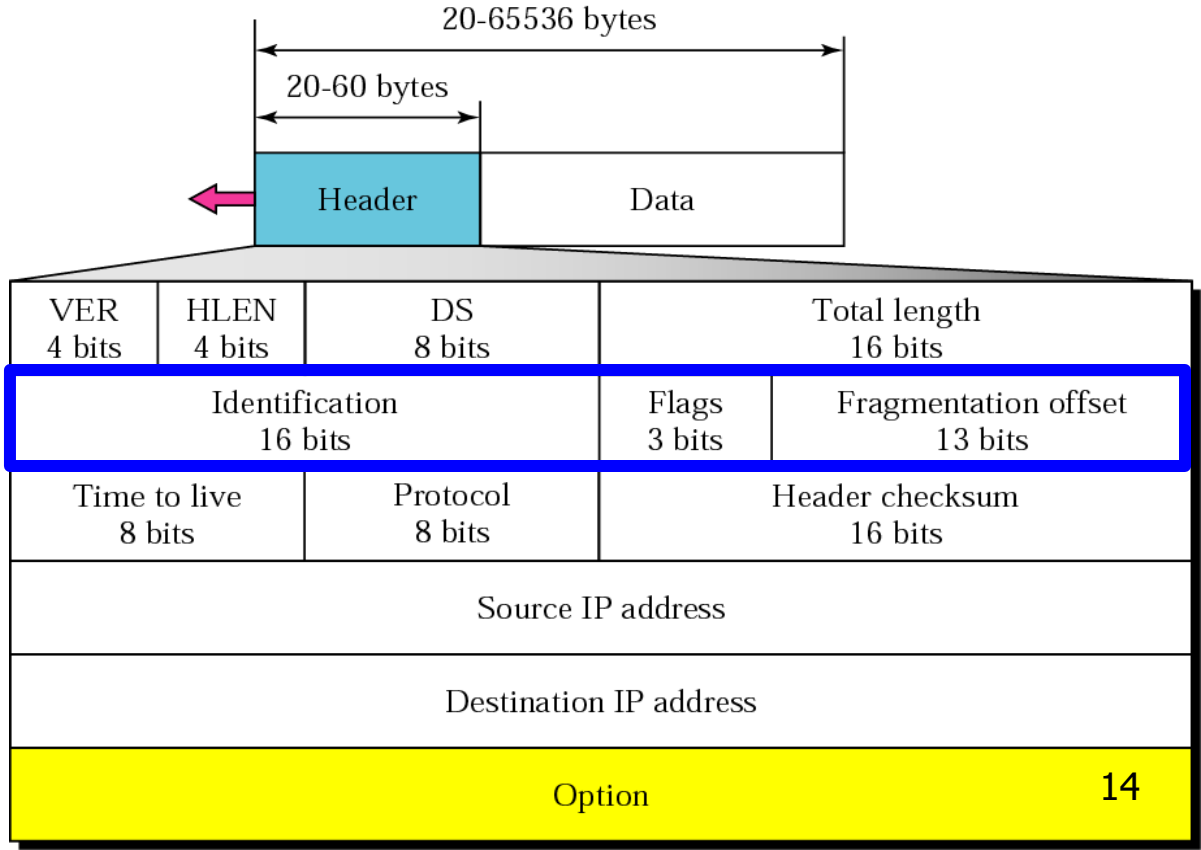
- 16 bit field defines total length of **header + data** of datagram.
- To find length of actual data, subtract header length from this.
- Important field w.r.t. fragmentation.



IPv4 datagram Format

Identification, Flags, Fragmentation Offset

- Used in case of fragmentation.

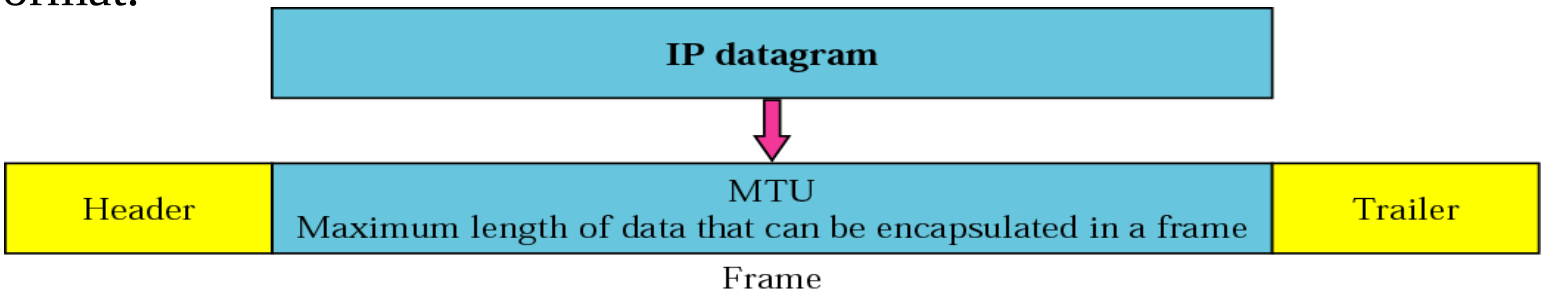


Fragmentation

- A datagram travels through different networks. Each router decapsulates IP datagram from frame, processes on it & then encapsulates in different frame.
- **The format and size of a frame depends on the protocol** used by the **physical network**.
- A datagram may have to be fragmented to fit the protocol regulations.
- Datagram is fragmented by **source host** or any **router** on path.
- Fragmented datagrams **may take different path**.
- **Re-assembling** of fragmented datagram will be done at the **final destination**

Fragmentation – Maximum Transfer Unit (MTU)

- Every Data Link Layer protocol has different frame format.



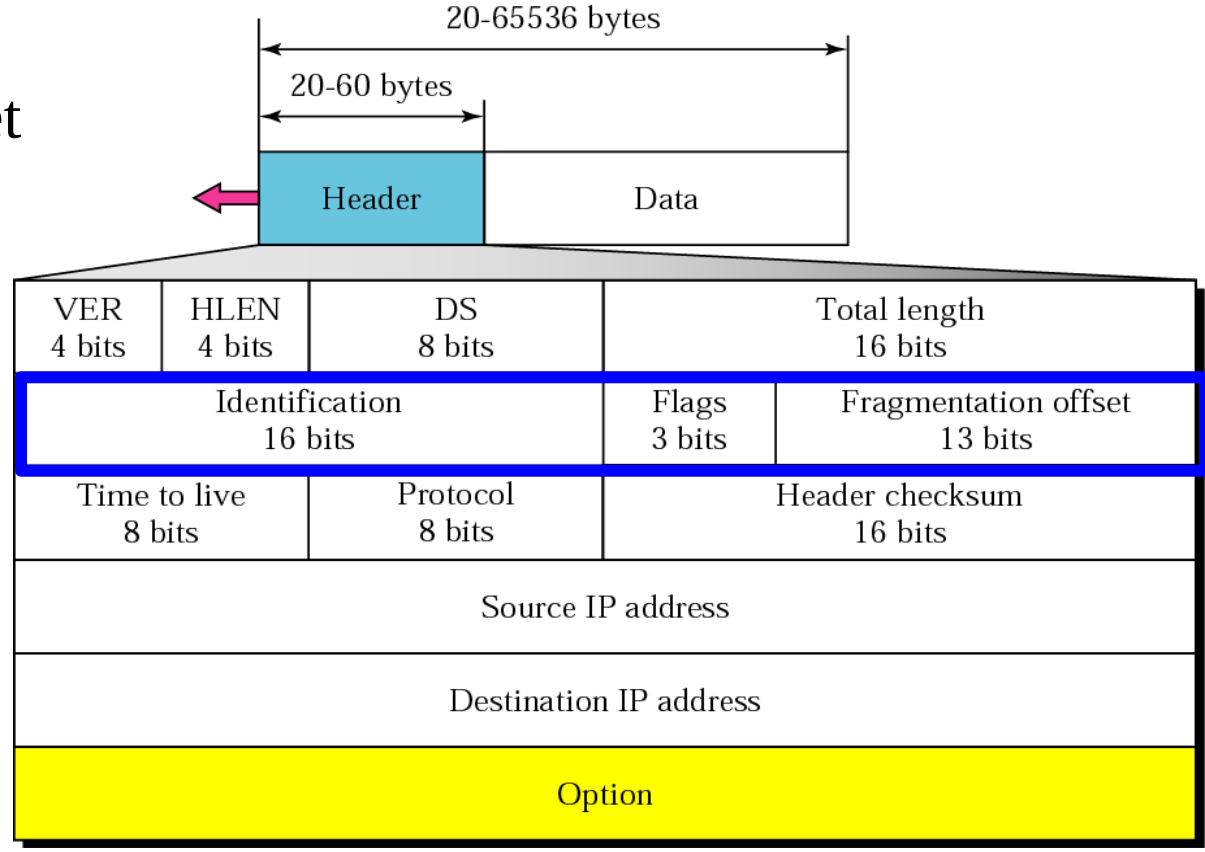
- Value of MTU differs from one physical n/w protocol to another.
- Max length of IP datagram is 65,535 bytes.
- If MTU of a physical n/w is less than this, then divide IP datagram. Called as **fragmentation**.

Protocol	*MTU*
Hyperchannel	65,535
Token Ring (16 Mbps)	17,914
Token Ring (4 Mbps)	4,464
FDDI	4,352
Ethernet	1,500
X.25	576
PPP	296

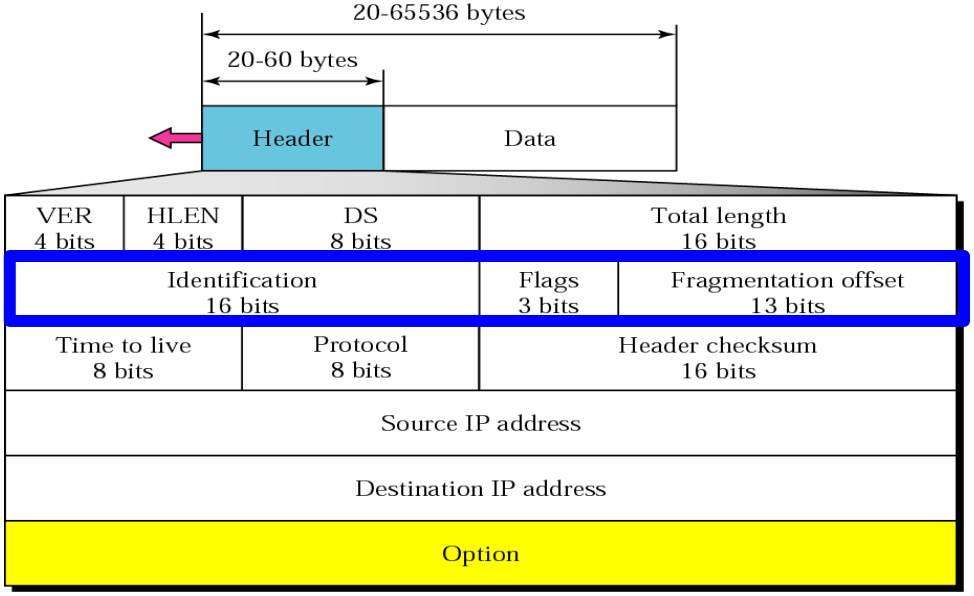
Fragmentation

Fields contributing to fragmentation are

- Identification
- Flags
- Fragmentation Offset

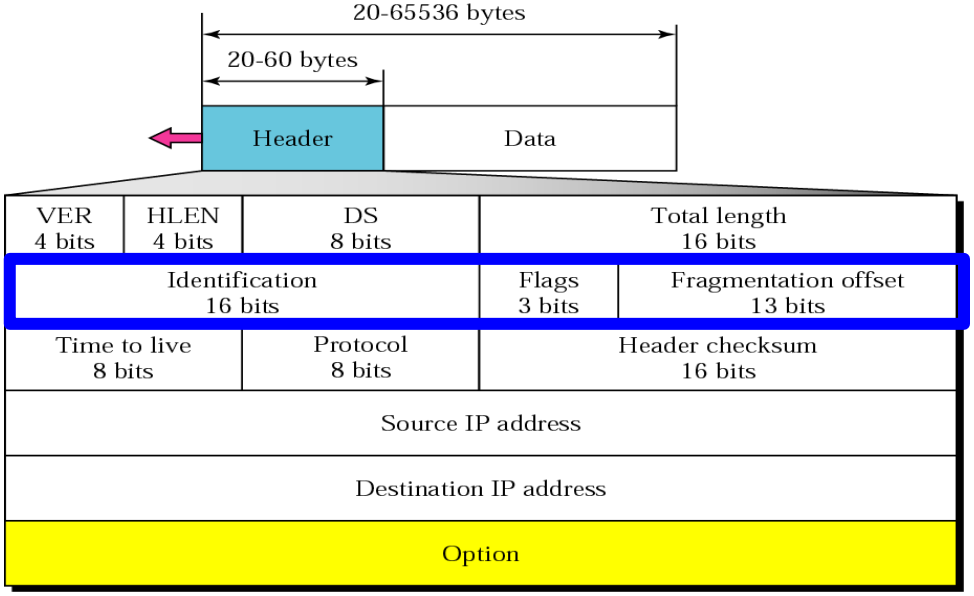


Identification field



- Combination of *identification number + source IP address*
- Identification number is generated using **counter** scheme.
- Counter = positive number, every time for new datagram, counter value is incremented.
- When **datagram is fragmented**, **identification field** is **copied** as it is from the original datagram to all fragmented datagram.
- This will help in reassembling it back.

Flags field

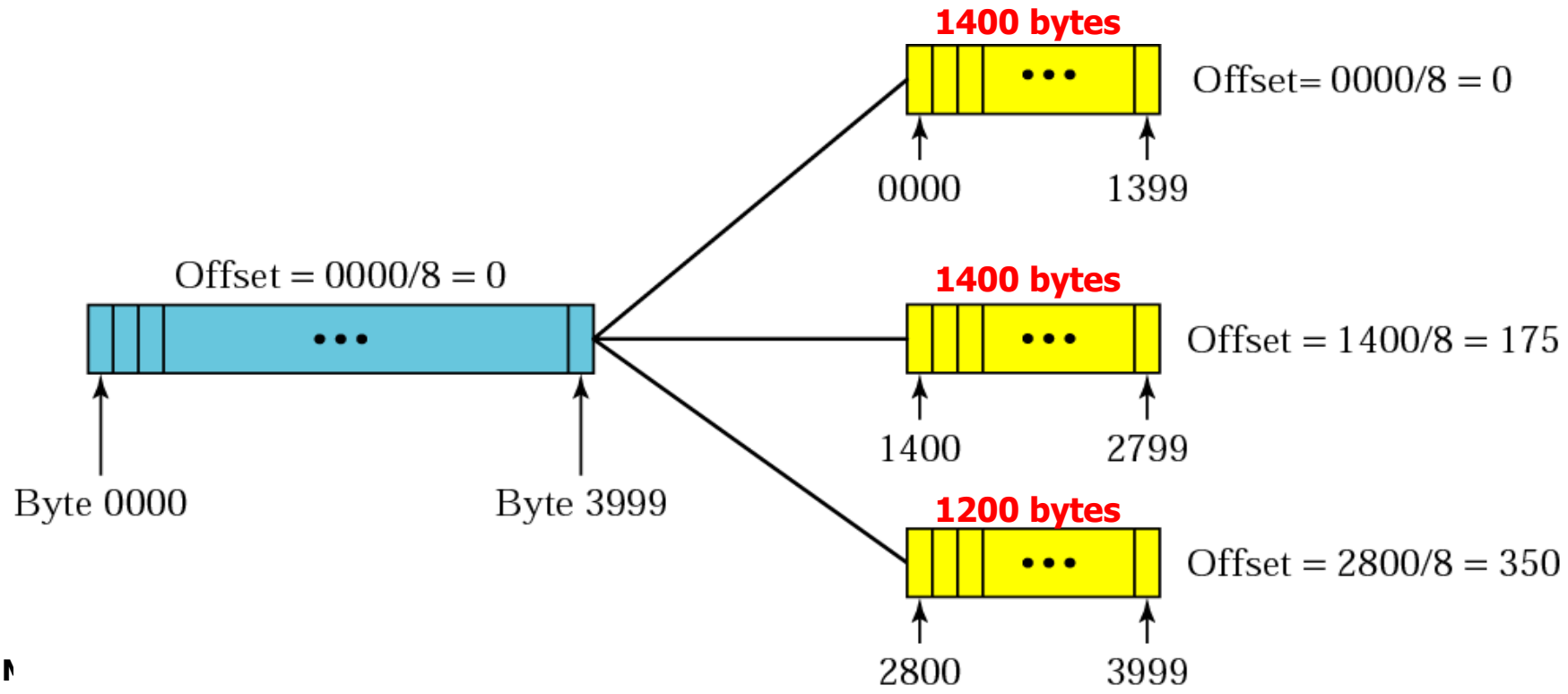


- 3 bit field
- 1st is reserved
- 2nd bit – if set to 1, DO NOT FRAGMENT
  if set to 0 , can be fragmented
- 3rd bit – if set to 1, more sub datagram of the same datagram exists
  if set to 0, that is the last fragment of the datagram.

D: Do not fragment
M: More fragments

Fragmentation Offset example

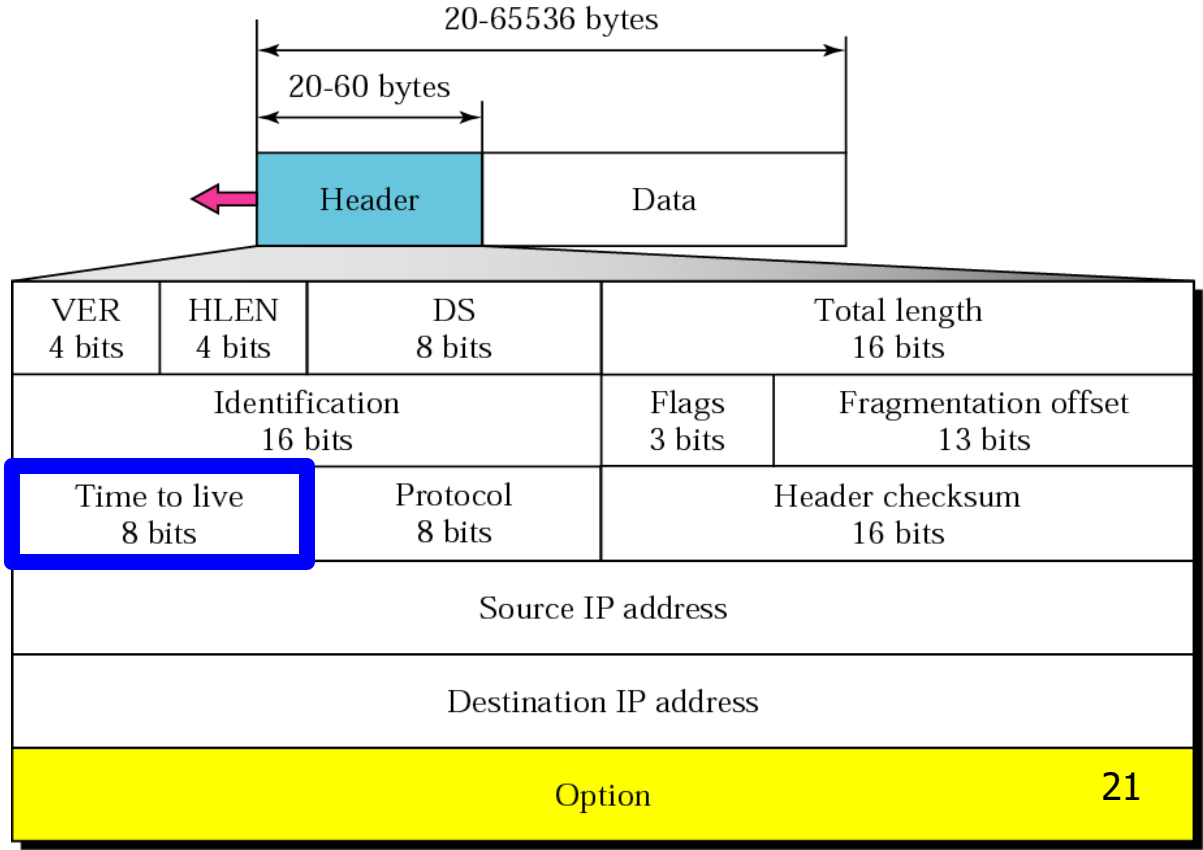
- 13 bit field , shows relative position of the fragment w.r.t original datagram



IPv4 datagram Format

Time to live

- Stores life time of a datagram.



IPv4 datagram Format

Time to live

- Field is used for **controlling maximum hops(routers) visited by datagram**.
- Source host puts the initial value as = 2 * (maximum number of hops between 2 host)
- On receiving of datagram, every router decrements this value by 1 and **check** if the **new value is 0**. If yes, discard the datagram.
- The reason is '**The routing table in the internet may be corrupted**' as a result the datagram is passed between same set of routers for a long time.

IPv4 datagram Format

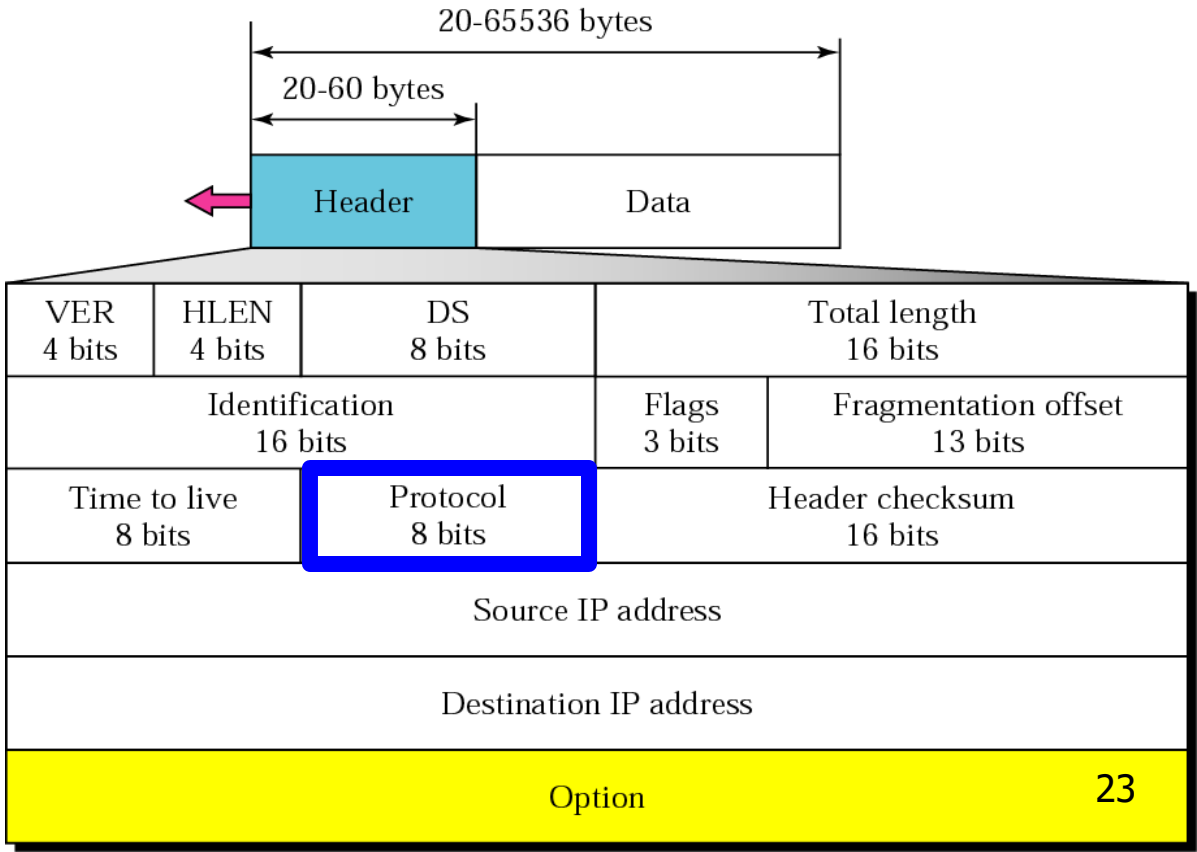
Time to live

- Stores life time of a datagram.

Protocol

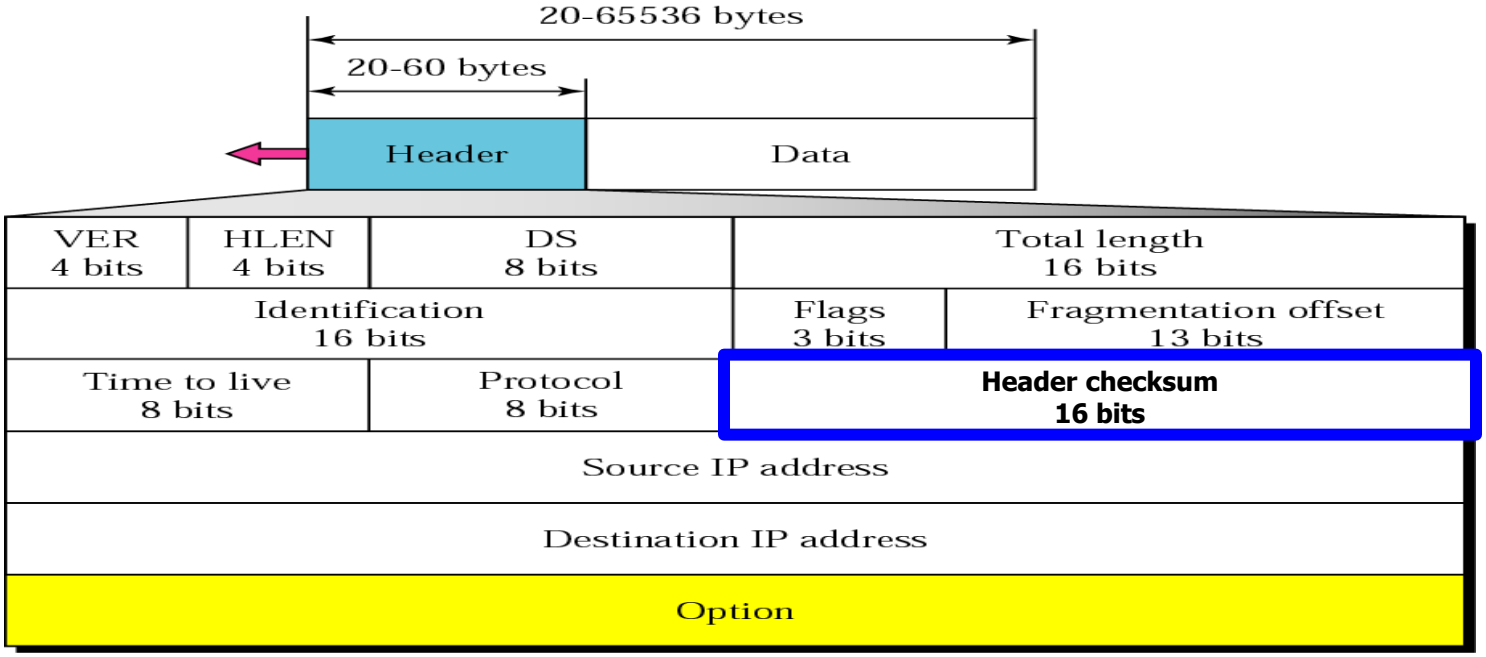
- Defines protocol used by IP layer.

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF



Checksum

- **Error detection** method.
- Calculated at sender & value is sent to receiver.
- Receiver repeats the same calculation
- If result is satisfactory, packet is accepted

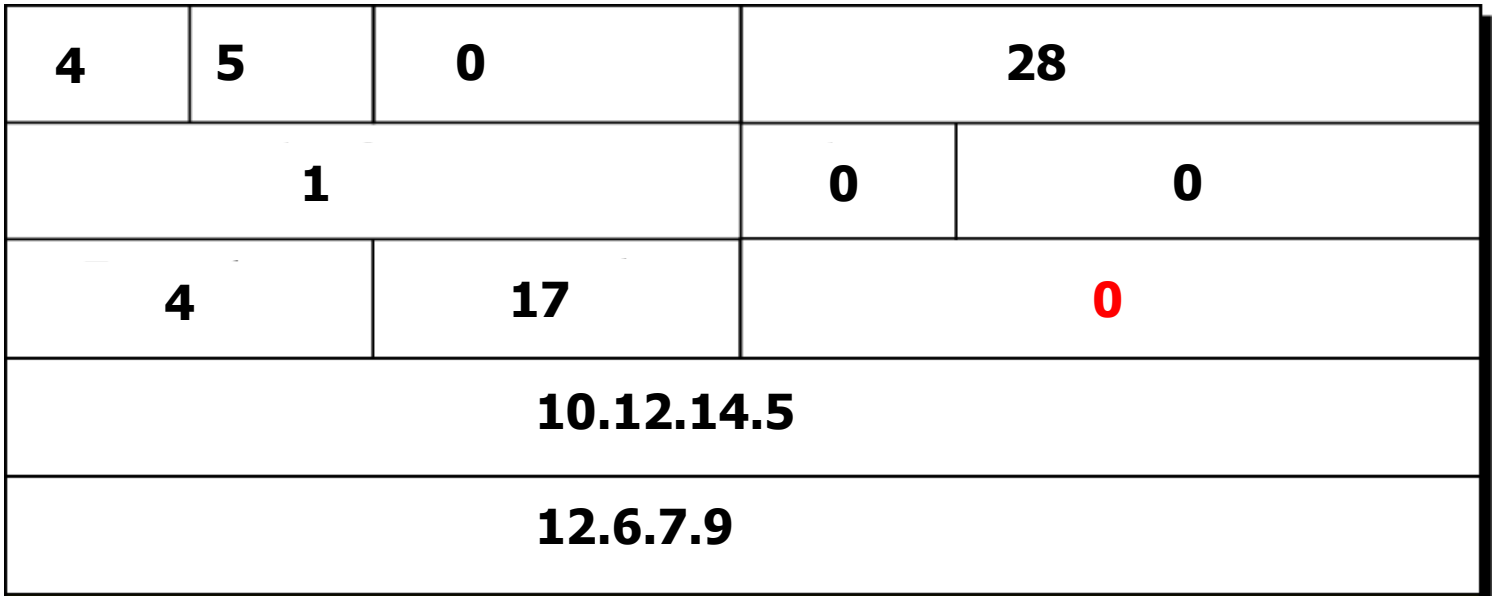


Checksum

- To create the checksum the sender does the following,
  - The packet is divided into k sections, each of n bits**. (n=16)**
  - All sections are added together using 1's complement arithmetic.
  - The final result is complemented to make the checksum.

- At receiver's side,
  - The packet is divided into k sections.
  - Add all sections.
  - Complement the result. If **result is 0, packet is accepted** else rejected.

Checksum - Example



To create the checksum the **sender** does the following,

1. The packet is divided into k sections, each of n bits. (n=16)
2. All sections are added together using 1's complement arithmetic.
3. The final result is complemented to make the checksum.
4. Substitute the final result for 0.

Checksum - Example

4,5 and 0 →	01000101	00000000
28 →	00000000	00011100
1 →	00000000	00000001
0 and 0 →	00000000	00000000
4 and 17 →	00000100	00010001
0 →	00000000	00000000
10.12 →	00001010	00001100
14.5 →	00001110	00000101
12.6 →	00001100	00000110
7.9	00000111	00001001
Sum with 1's complement	01110100	01001110
Checksum	10001011	10110001

35761

Checksum - Example

4	5	0	28
1		0	0
4		17	35761
10.12.14.5			
12.6.7.9			

At **receiver**'s side,

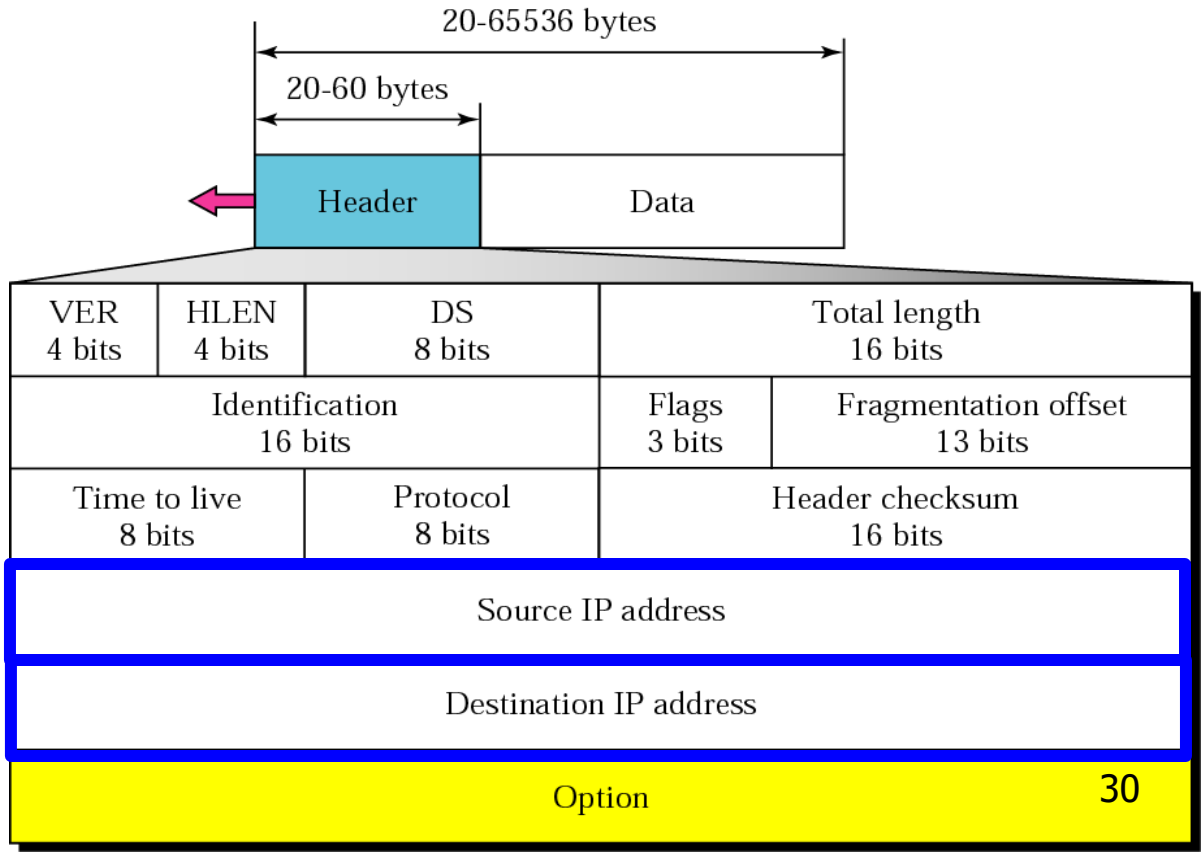
1. The packet is divided into k sections.
2. Add all sections.
3. Complement the result. If result is 0, packet is accepted else rejected.

Checksum - Example

4,5 and 0 →	01000101	00000000
28 →	00000000	00011100
1 →	00000000	00000001
0 and 0 →	00000000	00000000
4 and 17 →	00000100	00010001
35761 →	10001011	10110001
10.12 →	00001010	00001100
14.5 →	00001110	00000101
12.6 →	00001100	00000110
7.9	00000111	00001001
Sum with 1's complement	11111111	11111111
complement	00000000	00000000

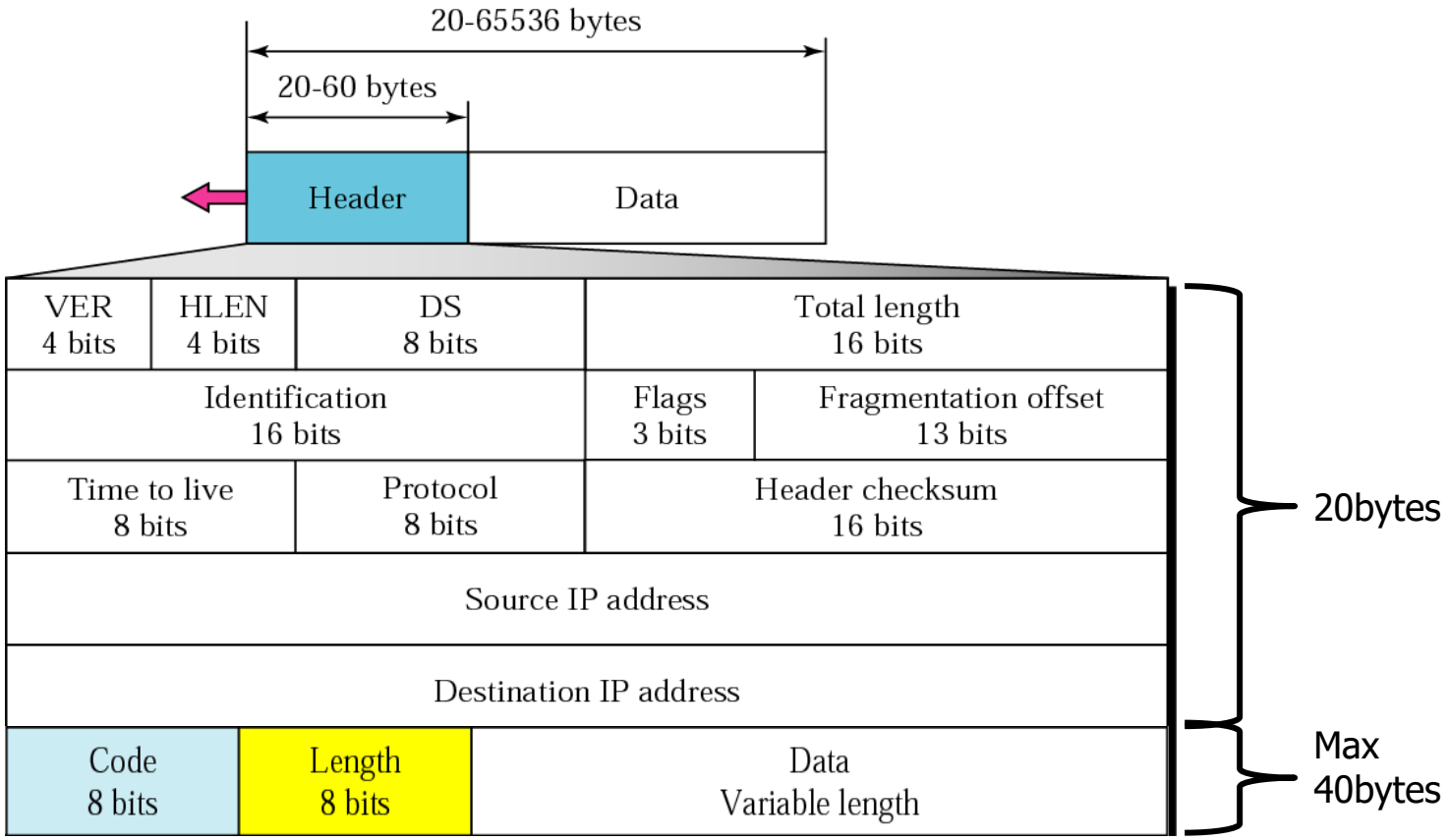
IPv4 datagram Format

Source Ip address and destination IP address



IPv4 datagram Format - Option field

- Variable part, can be a maximum of 40 bytes.
- Used for n/w testing and debugging

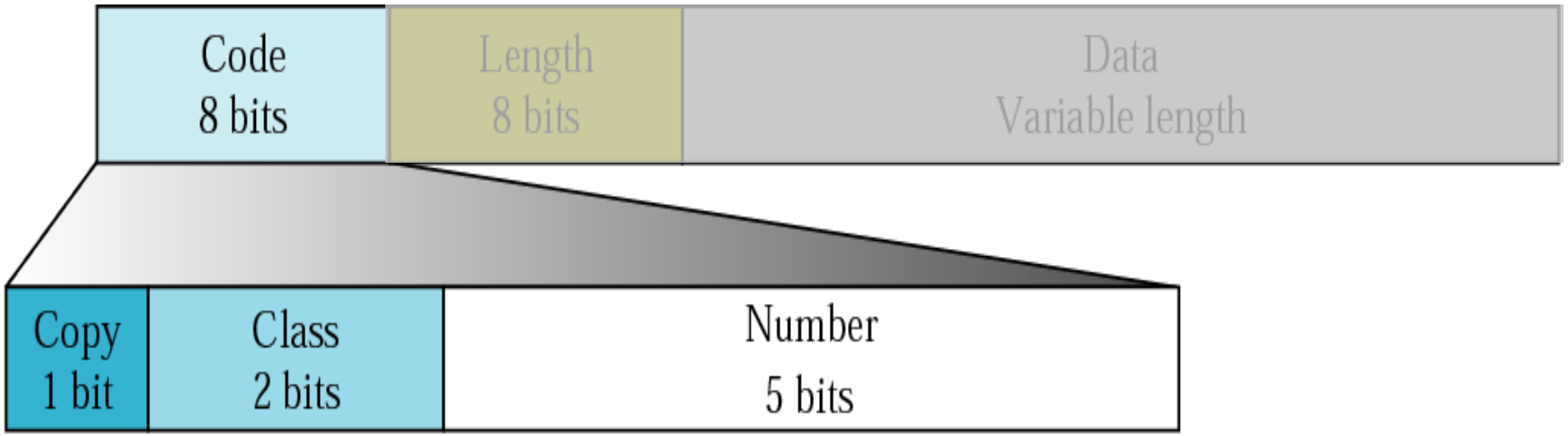


Option field –

- **Code** defines purpose of the option.
- **Length** defines **total length** of the option including the **code & length field.**
- **Data** field defines **data that specific option** requires.
- Code is present in all the options but remaining two fields are not present in all types of options.

Option field - Code

Class defines the general purpose of the option.
Number defines the type of option.



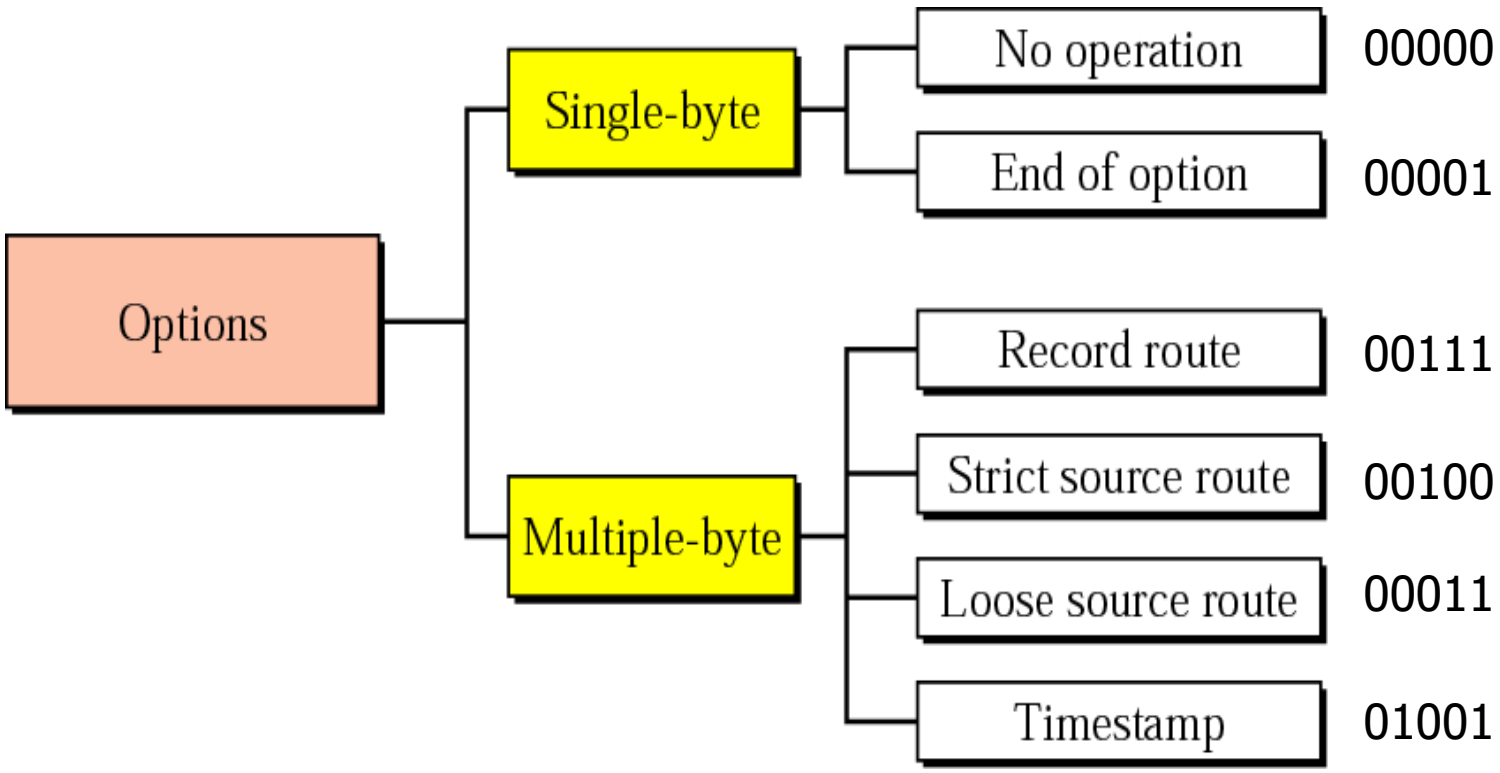
Copy
0 Copy only in first fragment
1 Copy into all fragments

Class
00 Datagram control
01 Reserved
10 Debugging and management
11 Reserved

Number
00000 End of option
00001 No operation
00011 Loose source route
00100 Timestamp
00111 Record route
01001 Strict source route

Option field – Categories of option

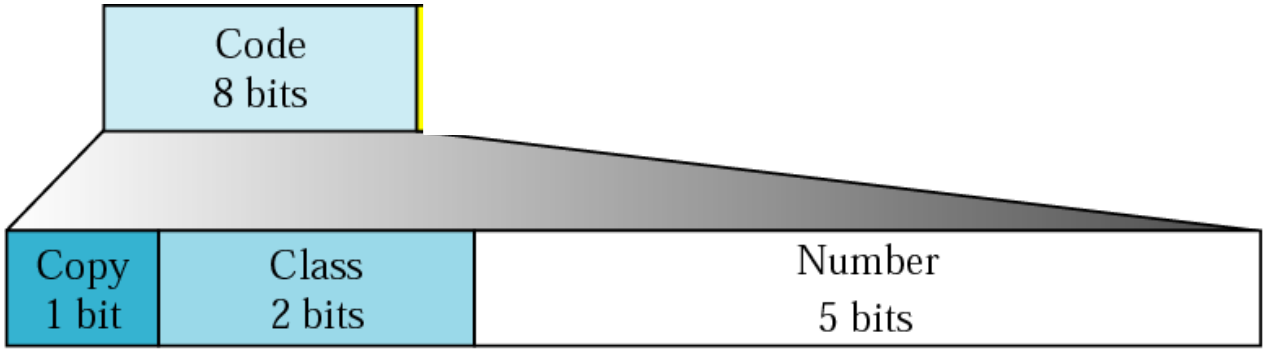
- Single byte options do not need length or data field.



Categories of option – No Operation Option

- Length & data fields are not present
- Used as a filler between two options.
  - Used to align next option.
  - Used to align beginning of option.

Code
00000001



Copy
0 Copy only in first fragment
1 Copy into all fragments

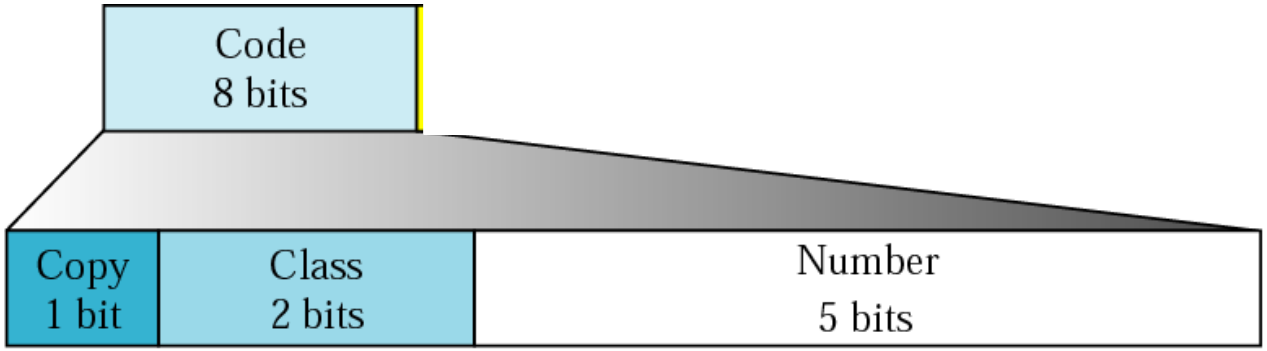
Class
00 Datagram control
01 Reserved
10 Debugging and management
11 Reserved

Number
00000 End of option
00001 No operation
00011 Loose source route
00100 Timestamp
00111 Record route
01001 Strict source route

Categories of option – End of option

- Length & data fields are not present.
- Used for padding.
- Used as the last option & only one *End of option* can be used

Code
00000000



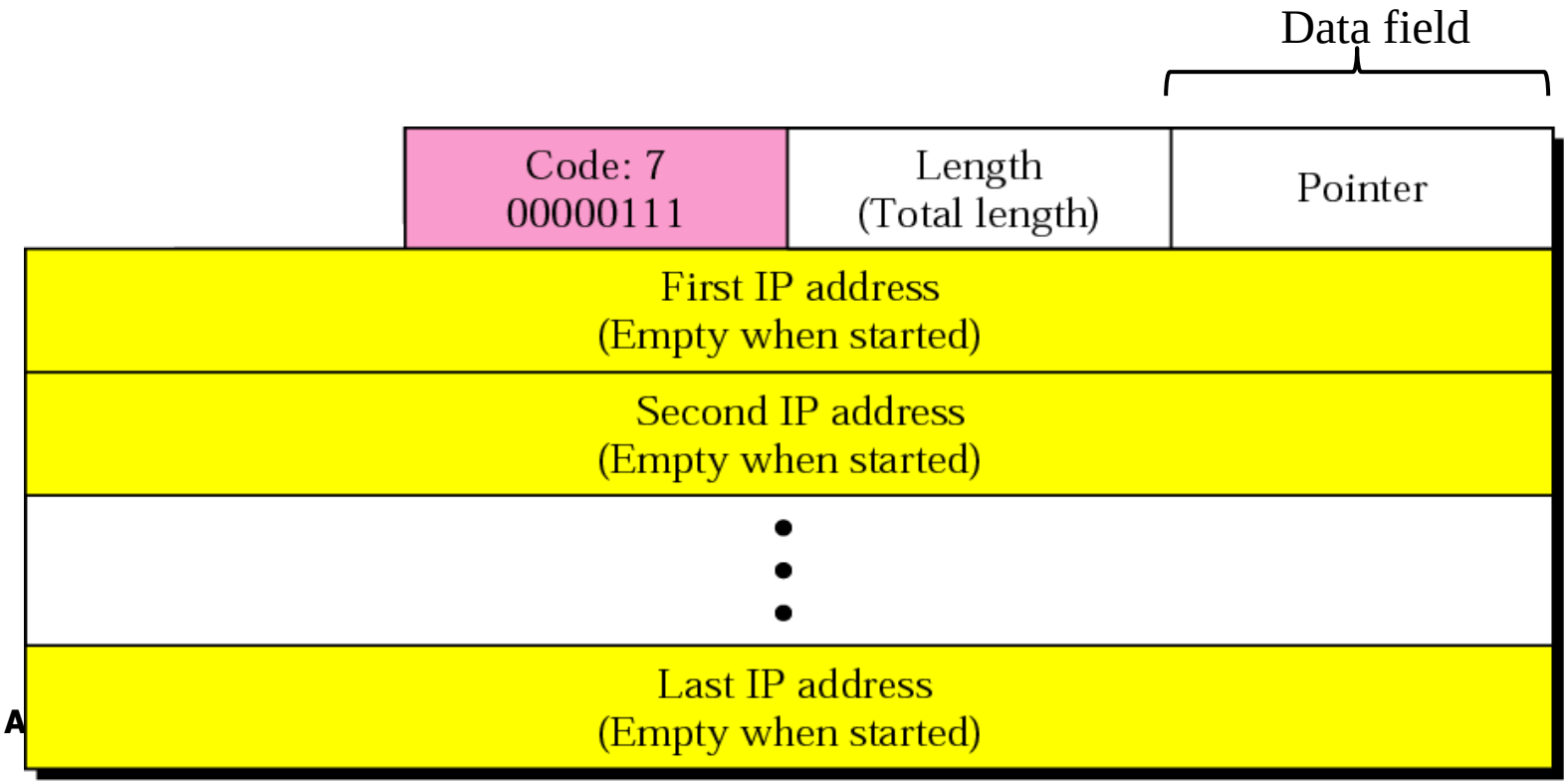
Copy
0 Copy only in first fragment
1 Copy into all fragments

Class
00 Datagram control
01 Reserved
10 Debugging and management
11 Reserved

Number
00000 End of option
00001 No operation
00011 Loose source route
00100 Timestamp
00111 Record route
01001 Strict source route

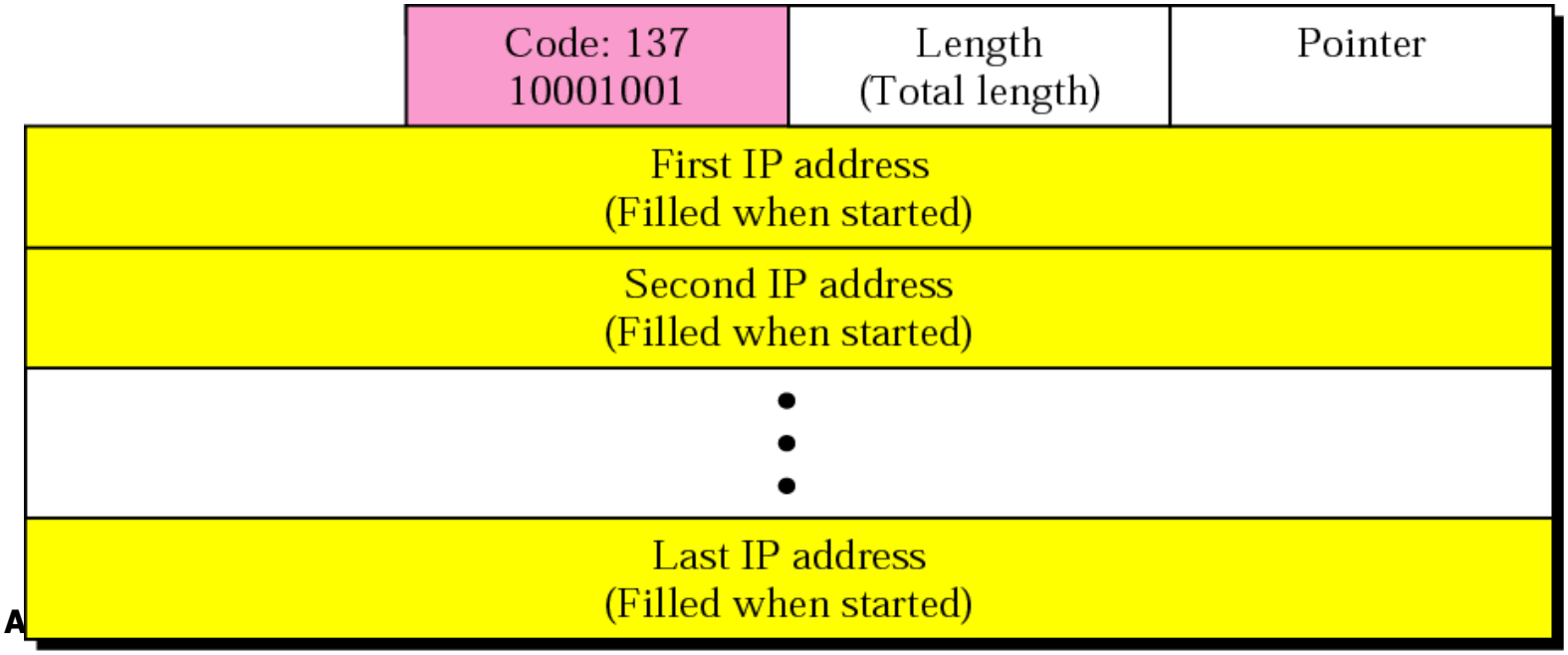
Categories of option – Record Route

- Length & data fields are present.
- **Used to record the internet routers** that handle the datagram
- Initially pointer field **points to the first available entry**.



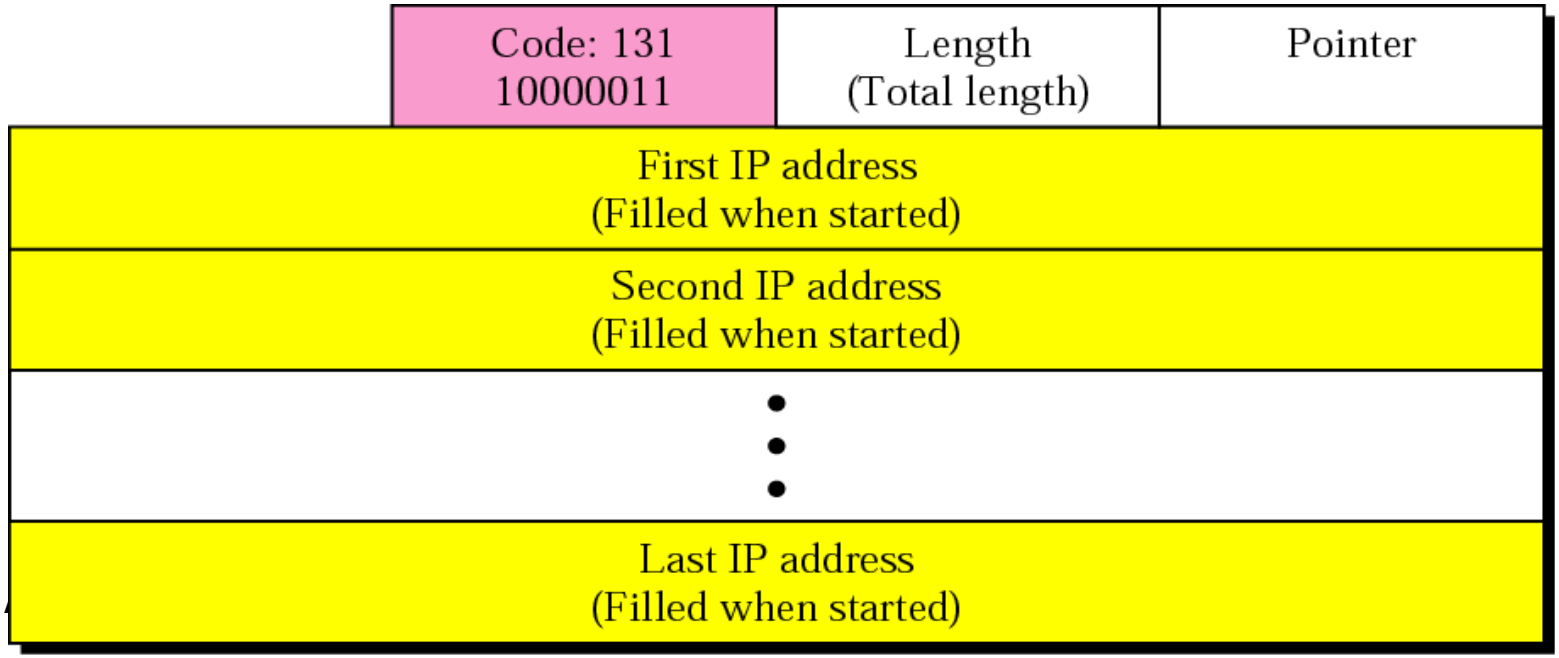
Categories of option – Strict Source Route Option

- Route is defined in advanced.
- Datagram **has to visit** all routers.
- Datagram is discarded and error is issued if
  - If datagram visits a router that is not listed

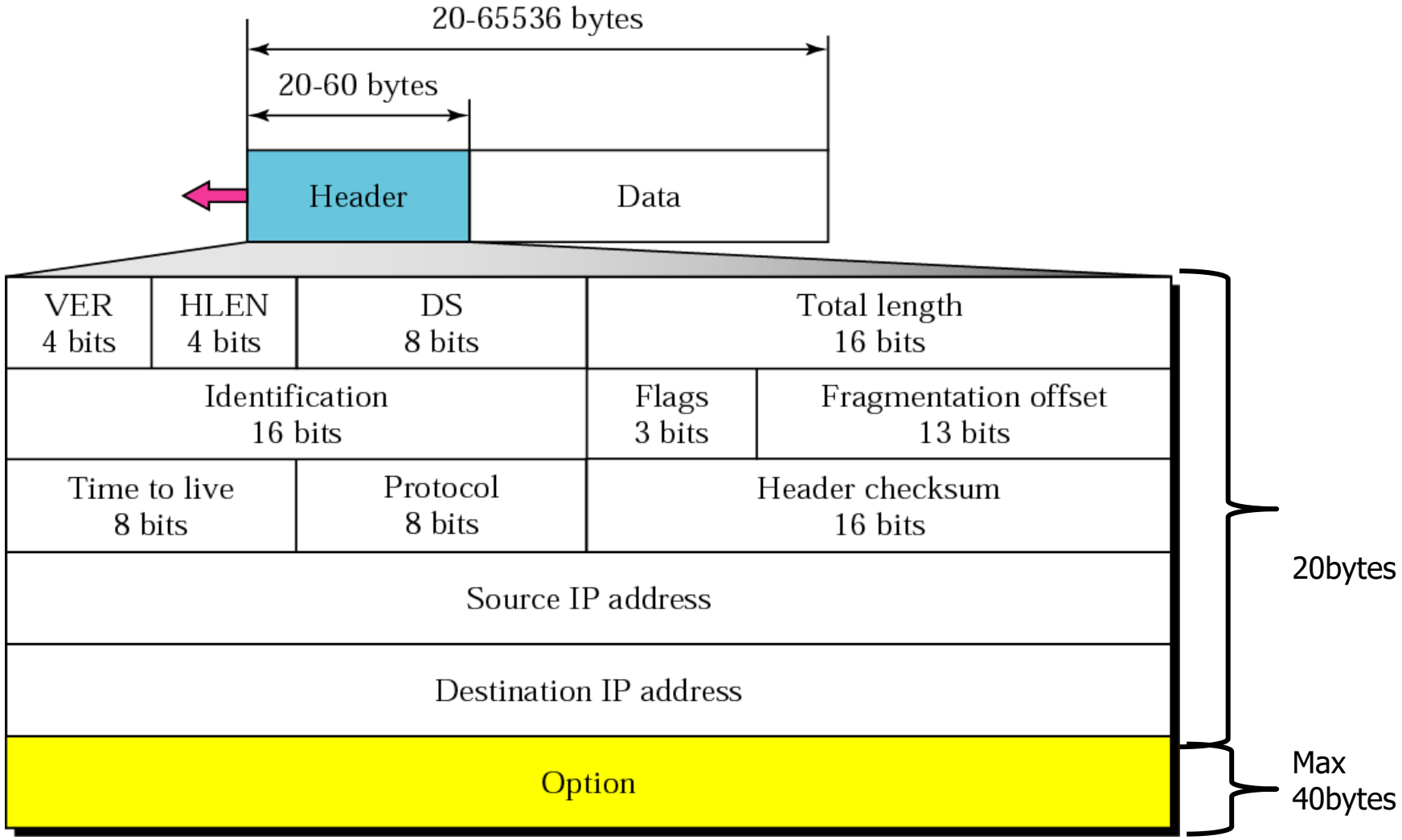


Categories of option – Loose Source Route Option

- Same as strict source route but it is **more relaxed**.
- Datagram **has to visit all routers** but it can visit other routers as well.

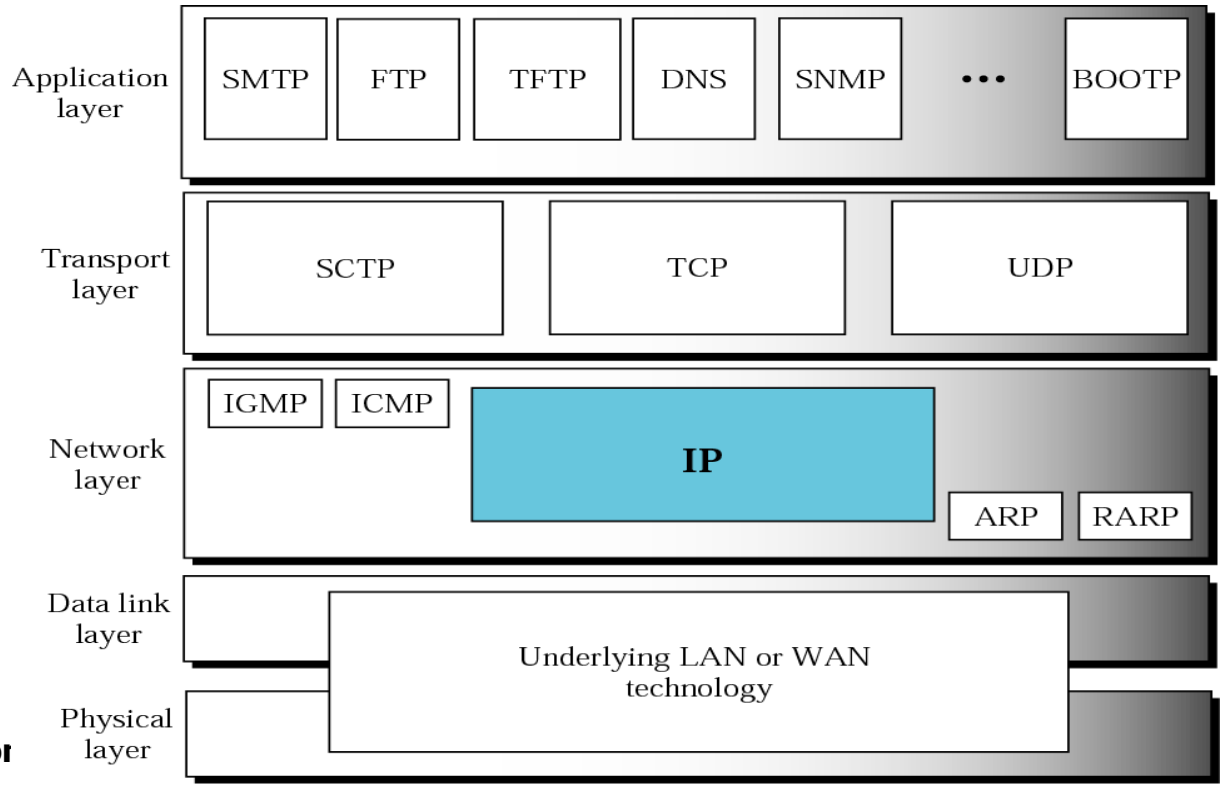


IPv4 datagram- Summary



Internet Control Message Protocol Version 4

ICMPv4



Basic Ideas – Need

- IP is unreliable. It has 2 deficiencies
  - Lack of **error control**.
  - Lack of **assistance mechanism**.

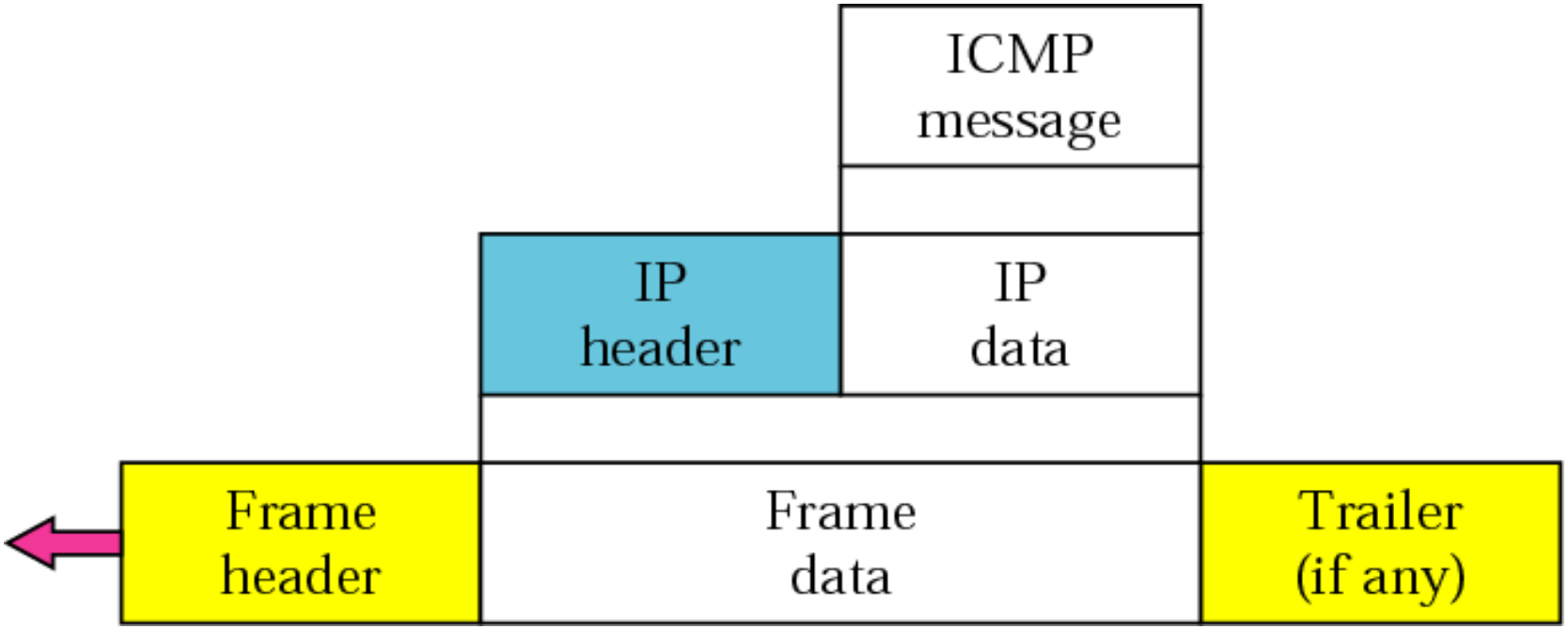
- ICMP compensates these two deficiencies.

- The Internet Control Message Protocol (ICMP) is a helper protocol that supports IP with facility for
  - **Error reporting**
  - **Simple queries**

- Function of ICMP
  - Generation of **ICMP messages**
  - ICMP **provides** some useful **diagnostics about network operation**

ICMP encapsulation

- ICMP is a network layer protocol.
- **ICMP messages** are not sent directly to DLL instead messages are **encapsulated inside IP datagram.**



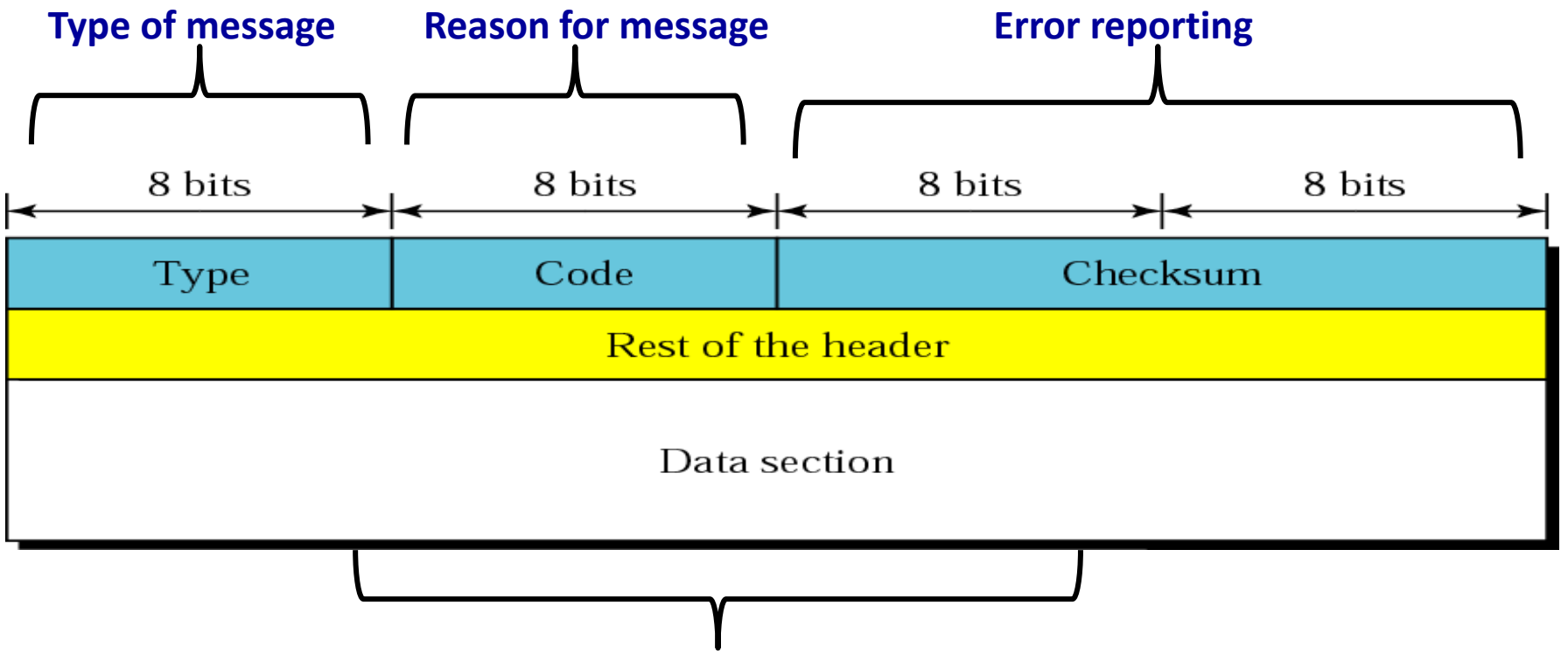
ICMP message

No ICMP error message will be generated in the following cases–

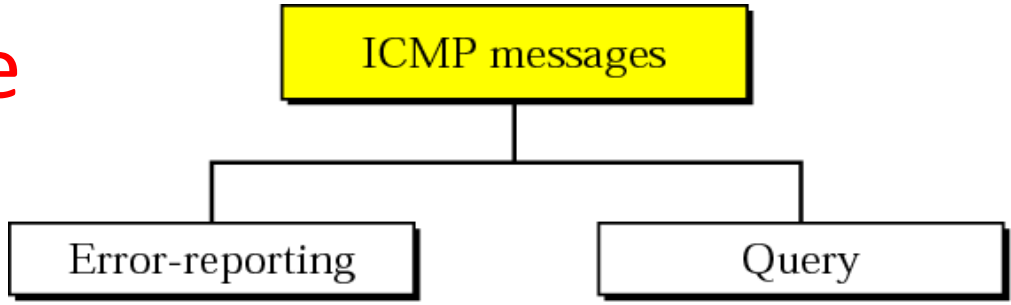
1. In response to a datagram **carrying an ICMP error message**.
2. For a fragmented datagram that is **not a first fragment**.
3. For a datagram **having multicast address**.
4. For a datagram having **special address** such as 127.0.0.0 or 0.0.0.0

Message Format

- 8-byte header and a variable-size data section.
- The general **format of the header** is **different for each message type**.
- The first 4 bytes are common to all.



Types of ICMP message



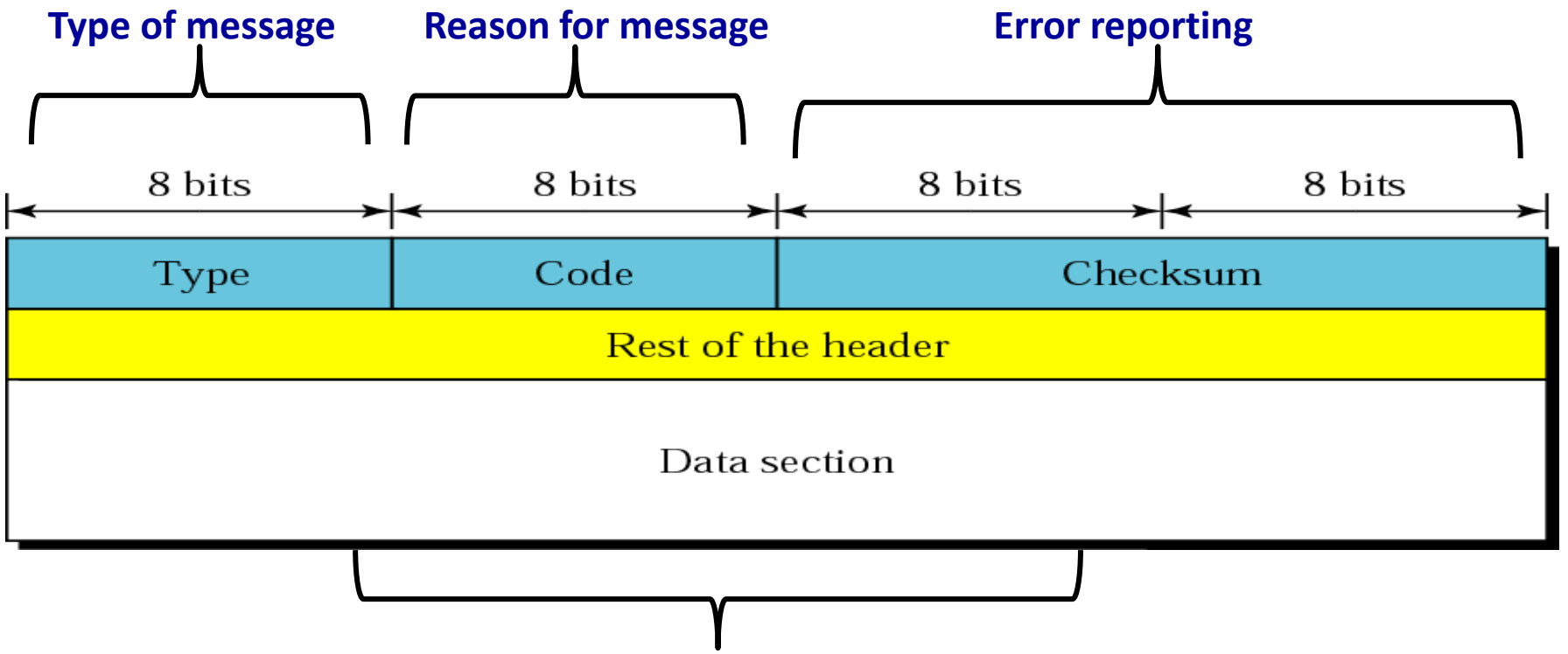
•The error-reporting messages **report problems** that a router or a host (destination) **may encounter**.

•The query messages **get specific information** from a router or another host.

Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

Message Format

- 8-byte header and a variable-size data section.
- The general **format of the header** is **different for each message type**.
- The first 4 bytes are common to all.



CHECKSUM

In ICMP the checksum is calculated over the entire message (header and data).

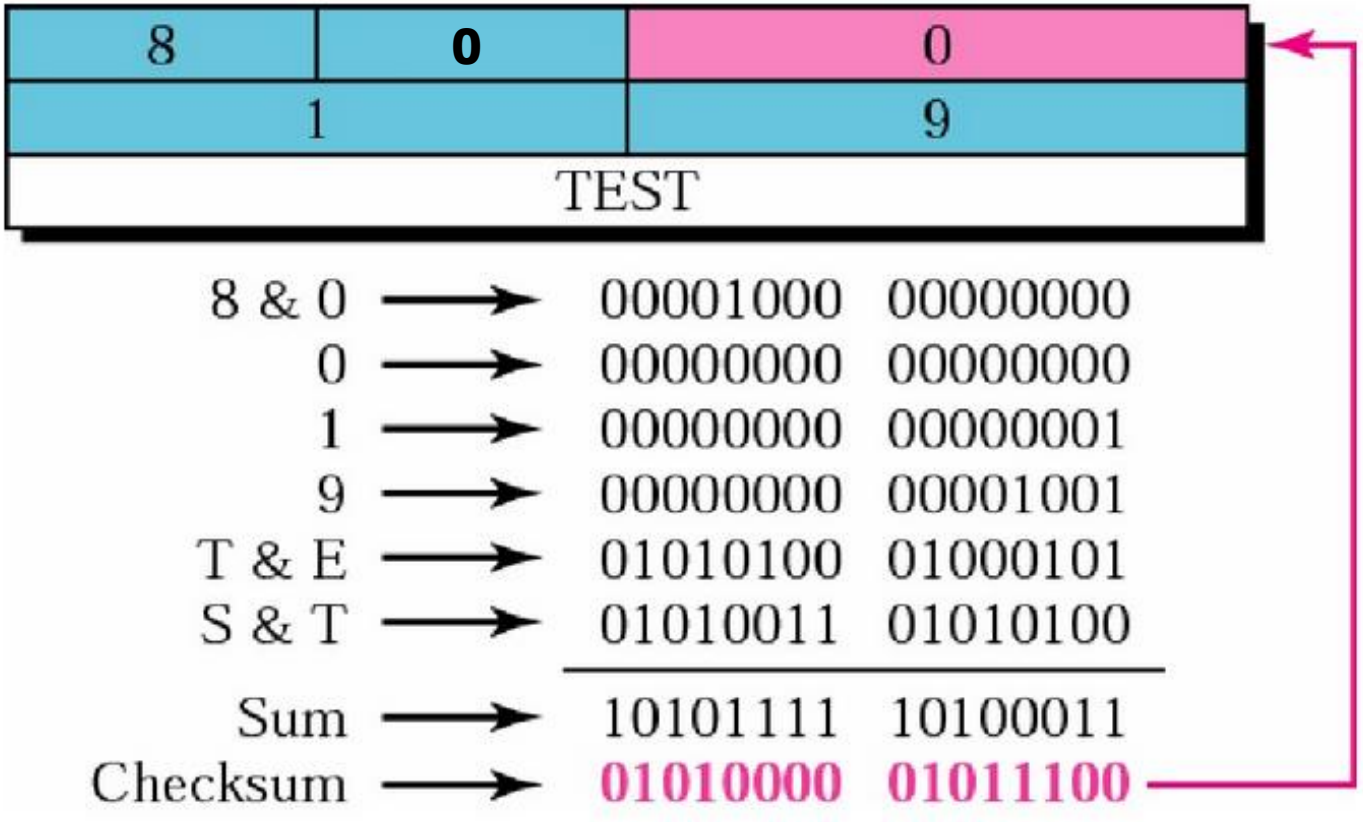
The sender follows these steps using 1's complement,

1. The checksum field is set to 0.
2. The sum of all 16 bits is calculated.
3. The sum is complemented to get the checksum.
4. The checksum is stored in the checksum field.

The receiver follows these steps using 1's complement,

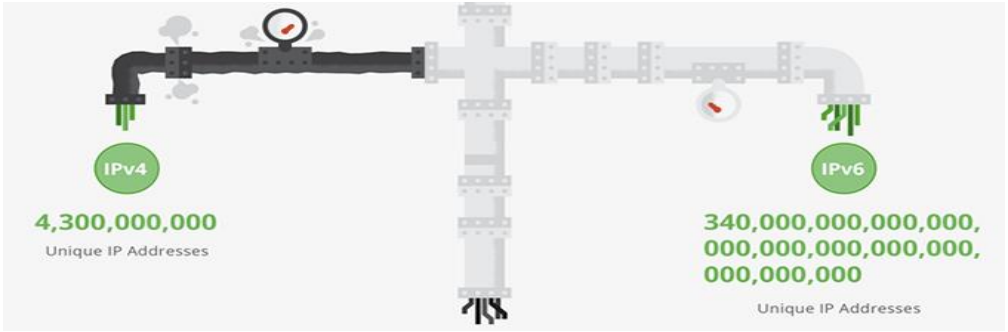
1. The sum of all words is calculated.
2. The sum is complemented.
3. If the result obtained in step 2 is 16 0s, the message is accepted , ow it is rejected.

CHECKSUM


8
0
0
1
9
TEST
8 & 0 → 00001000 00000000
0 → 00000000 00000000
1 → 00000000 00000001
9 → 00000000 00001001
T & E → 01010100 01000101
S & T → 01010011 01010100
Sum → 10101111 10100011
Checksum → 01010000 01011100

Internet Protocol IPv6 / IPng

Need for IPV6



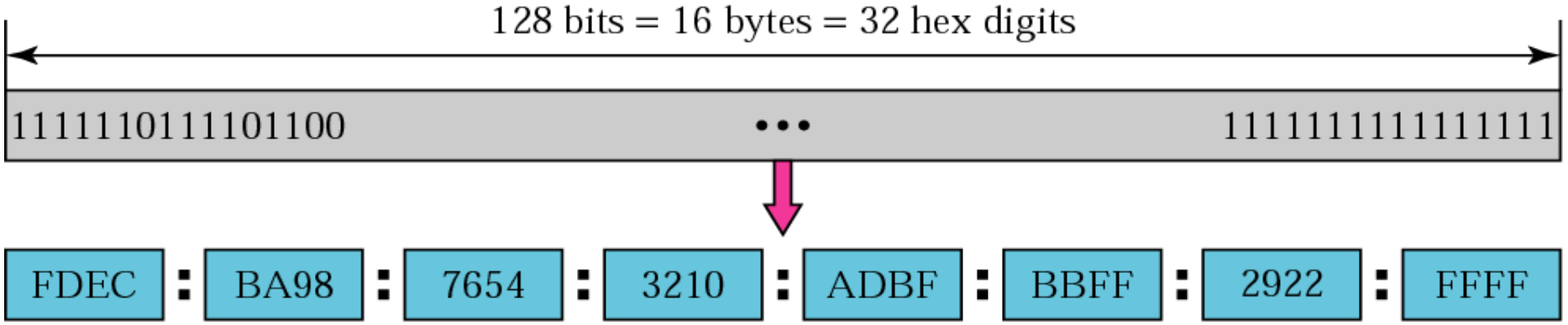
- IPv6 is the successor to IPv4.
- It functions similarly to IPv4 in that it provides the unique, numerical IP addresses necessary for Internet-enabled devices to communicate.
- IPv4 uses 32 bits for its Internet addresses. That means it can support **2^32 IP addresses** in total — around 4.29 billion.
- That may seem like a lot, but all 4.29 billion IP addresses have now been assigned to various institutions, leading to the crisis we face today.
- Major difference: it utilizes 128-bit addresses.
- IPv6 utilizes 128-bit Internet addresses. Therefore, it can support **2^128 Internet addresses** — 340,282,366,920,938,000,000,000,000,000,000,000,000 of them to be exact.

IPv6 - Advantages over IPv4

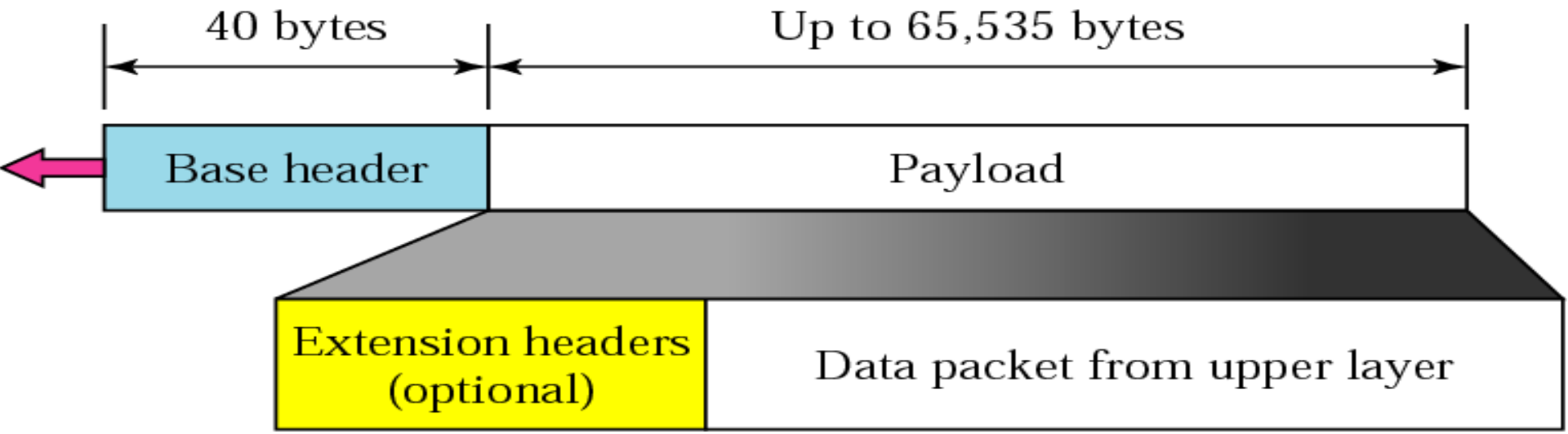
	IPv6	IPv4
Larger address space	128 bits long	32 bits long
Better header format	Option field is separated from header	Option field is with header only
New options	To add new functionality	-
Resource allocation	Support audio/video traffic. Instead of TOS, **flow label** and **traffic class** field is used	Only ***TOS*** is used
Security	Encryption / authentication option is introduced	No Such Option

IPv6 address

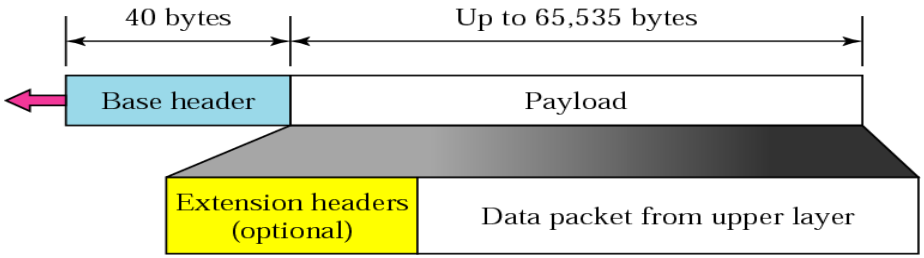
- To make address more readable, uses hexadecimal **colon notation**.
- 128 bits = 16bytes = **32hex** digits
- 128 bits divided into **8 sections**, each with 2 bytes
- 2 bytes require **4 hexadecimal** digits. (8*4=32 hexadigit)



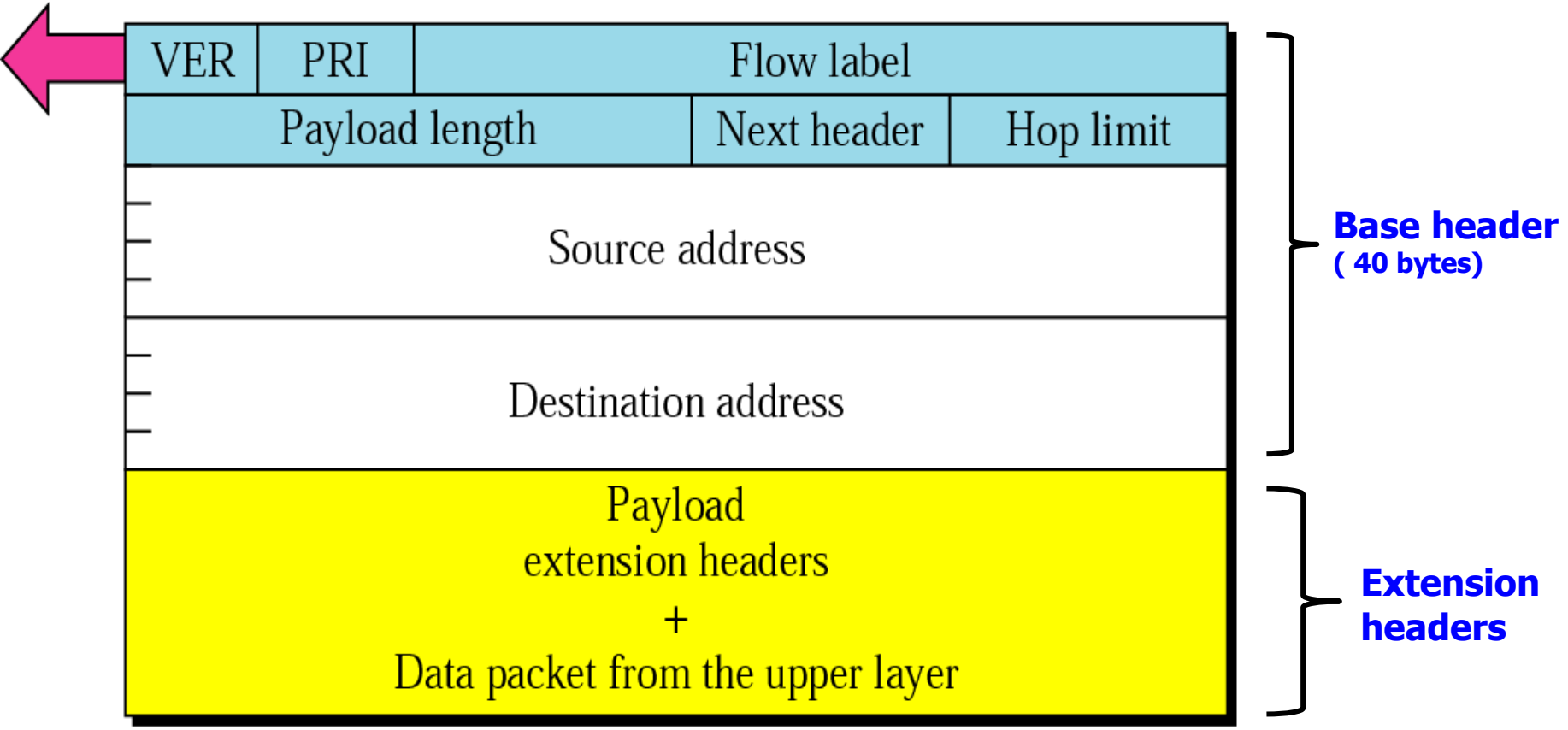
IPv6 datagram



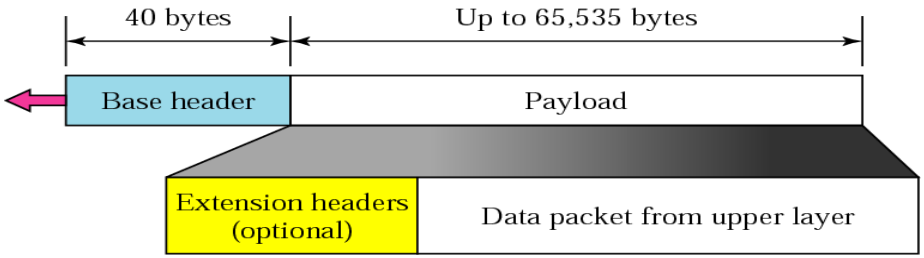
IPv6 datagram –



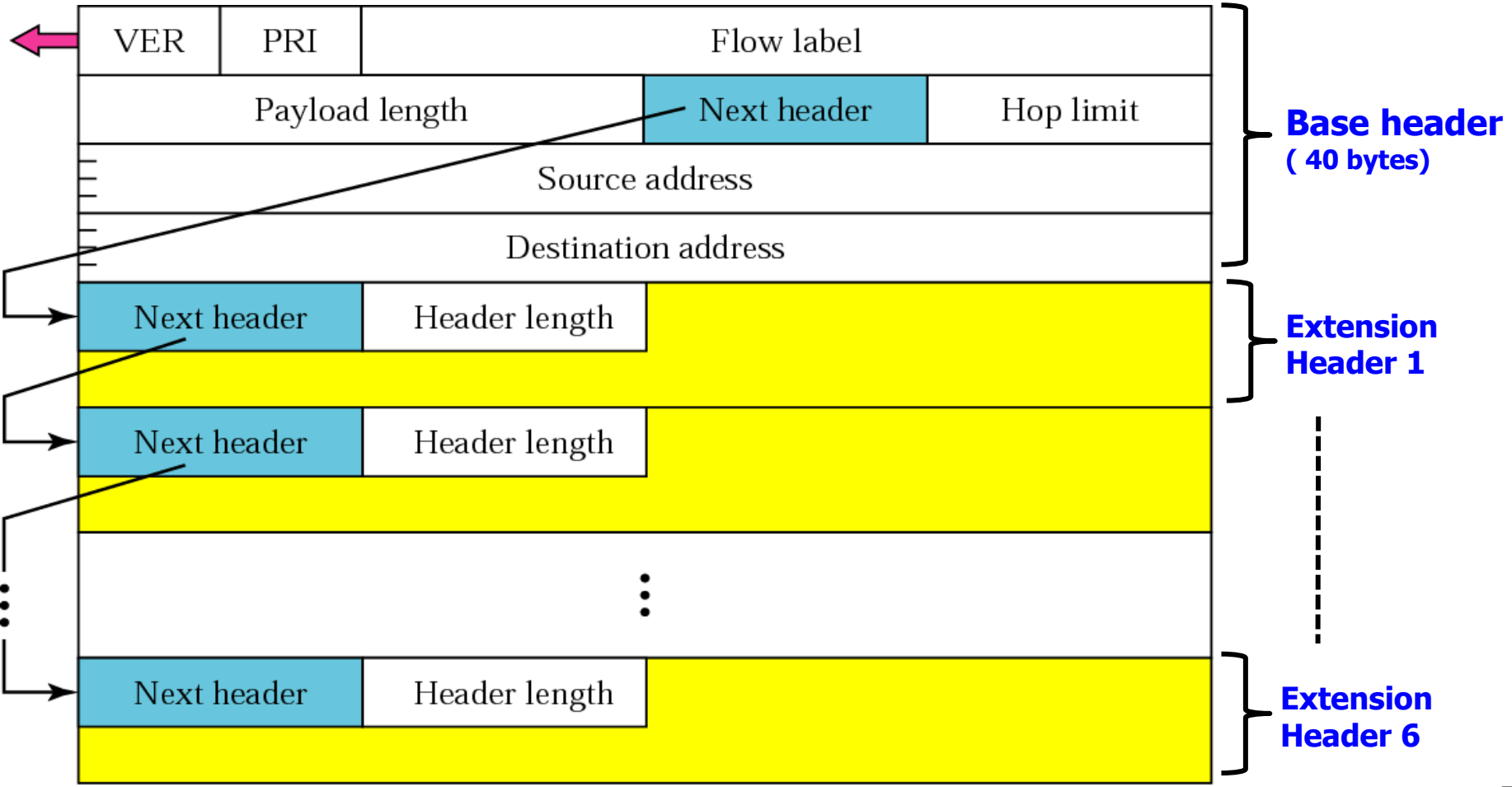
Base header & Extension header



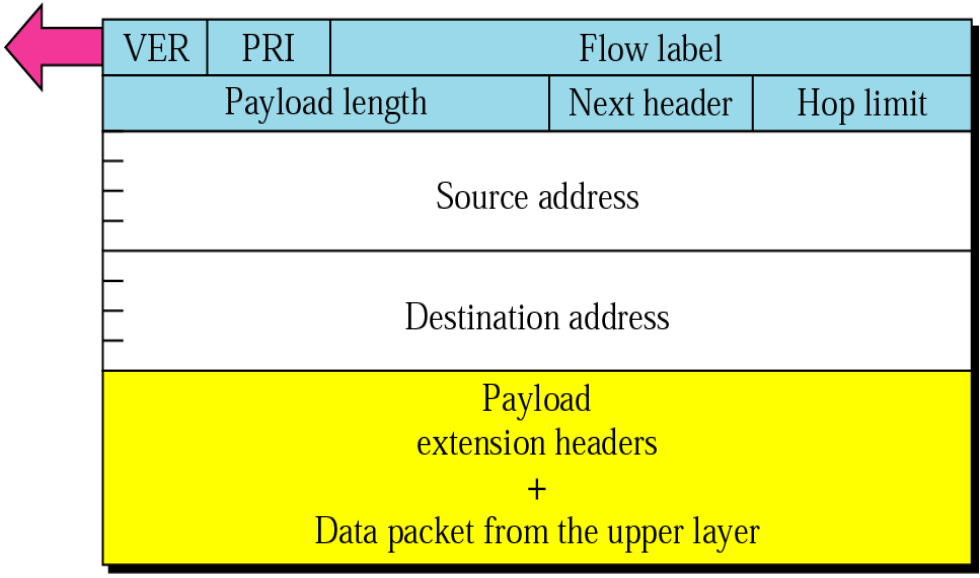
IPv6 datagram –



Base header & Extension header



IPv6 datagram

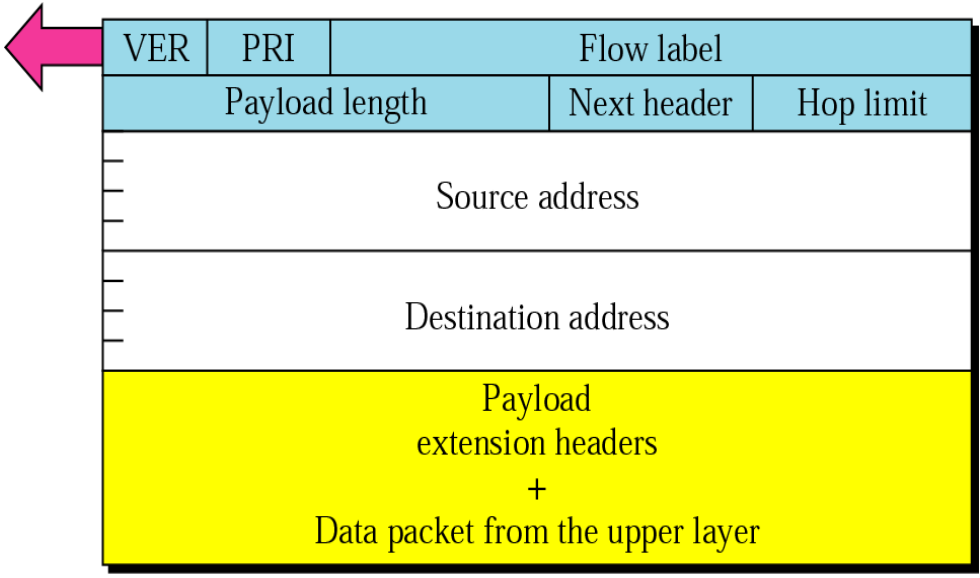


Header

- Version (4bits) : Version of IP e.g. 6
- Traffic class or Priority (8 bits): used to distinguish different payloads with different delivery requirement
- Flow Label(20bits) : Special field for flow control
- Payload Length (16bits) : Total length of datagram (excluding base header)
- Next Header (8 bits) : header that follows base header
- Hop limit (8 bits) : Same purpose as Time to live (TTL) field
- Source Address (16 byte (128bits)) : Sender's address
- Destination Address (16 byte (128bits)) : Receiver's address

Base header can be followed by optional Extension headers

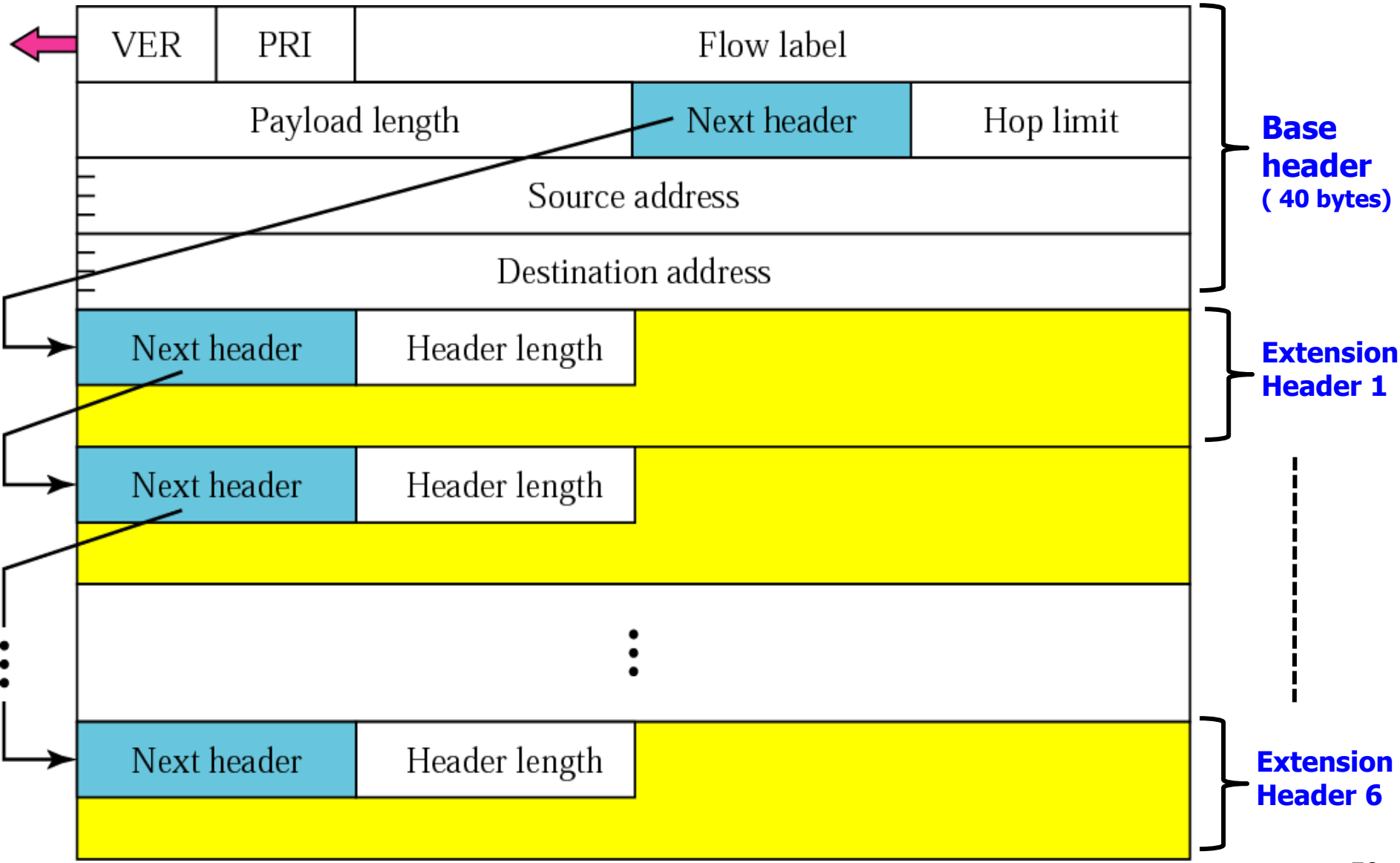
IPv6 datagram



Flow Label(20bits) :

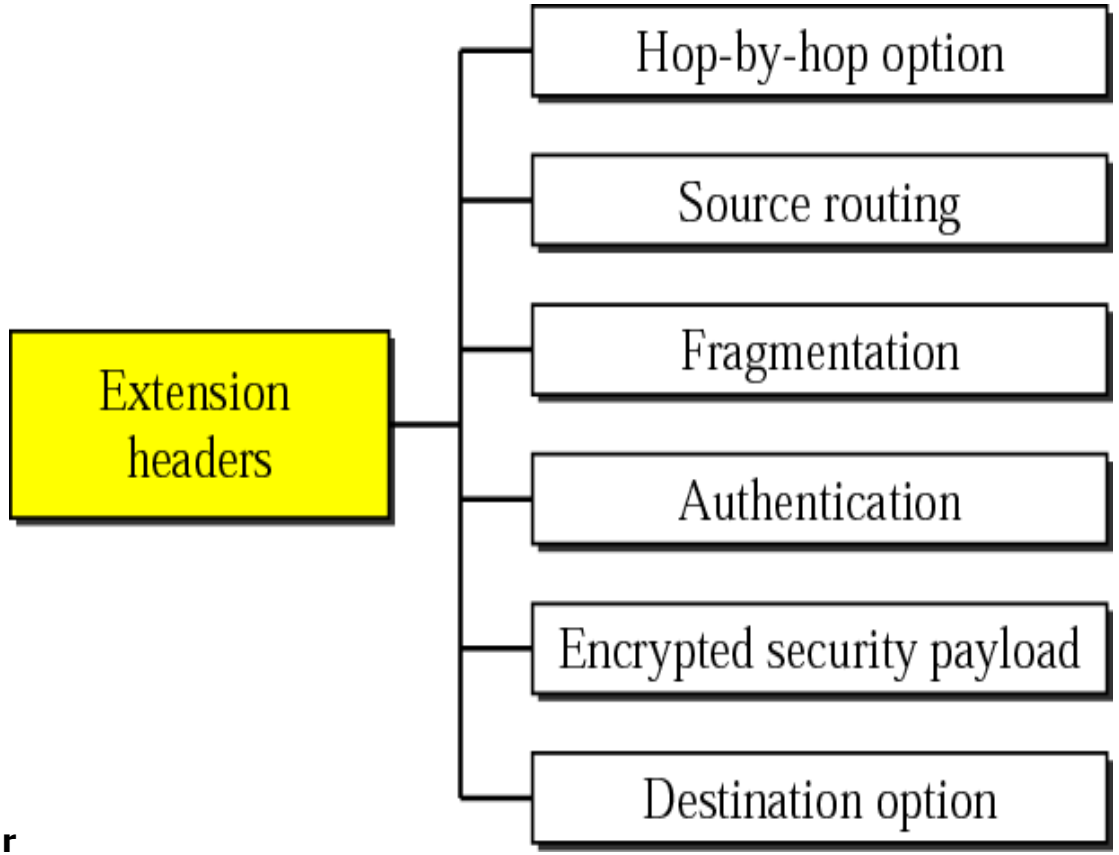
- It is same as that of ToS field in IPv4.
- A router that supports the handling of flow labels has a flow label table.
- The table has an entry for each active flow label; each entry defines the services required by the corresponding flow label.

IPv6 datagram – Extension header



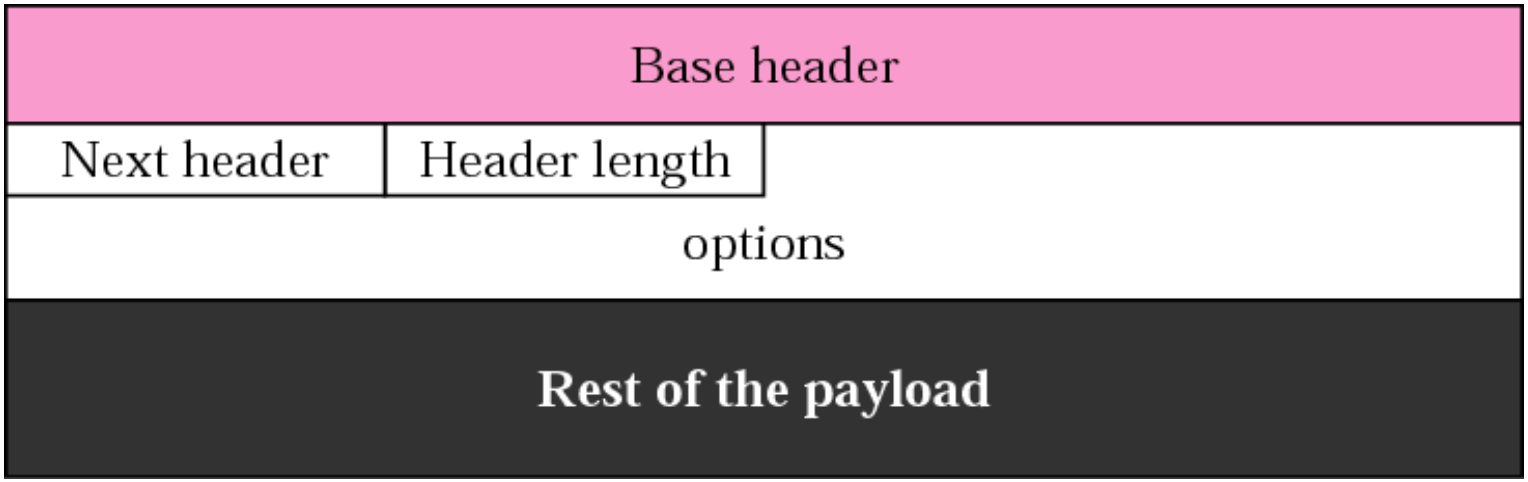
IPv6 datagram – Extension header

- Only present when needed.
- Eliminated IPv4's **40-byte limit on options**



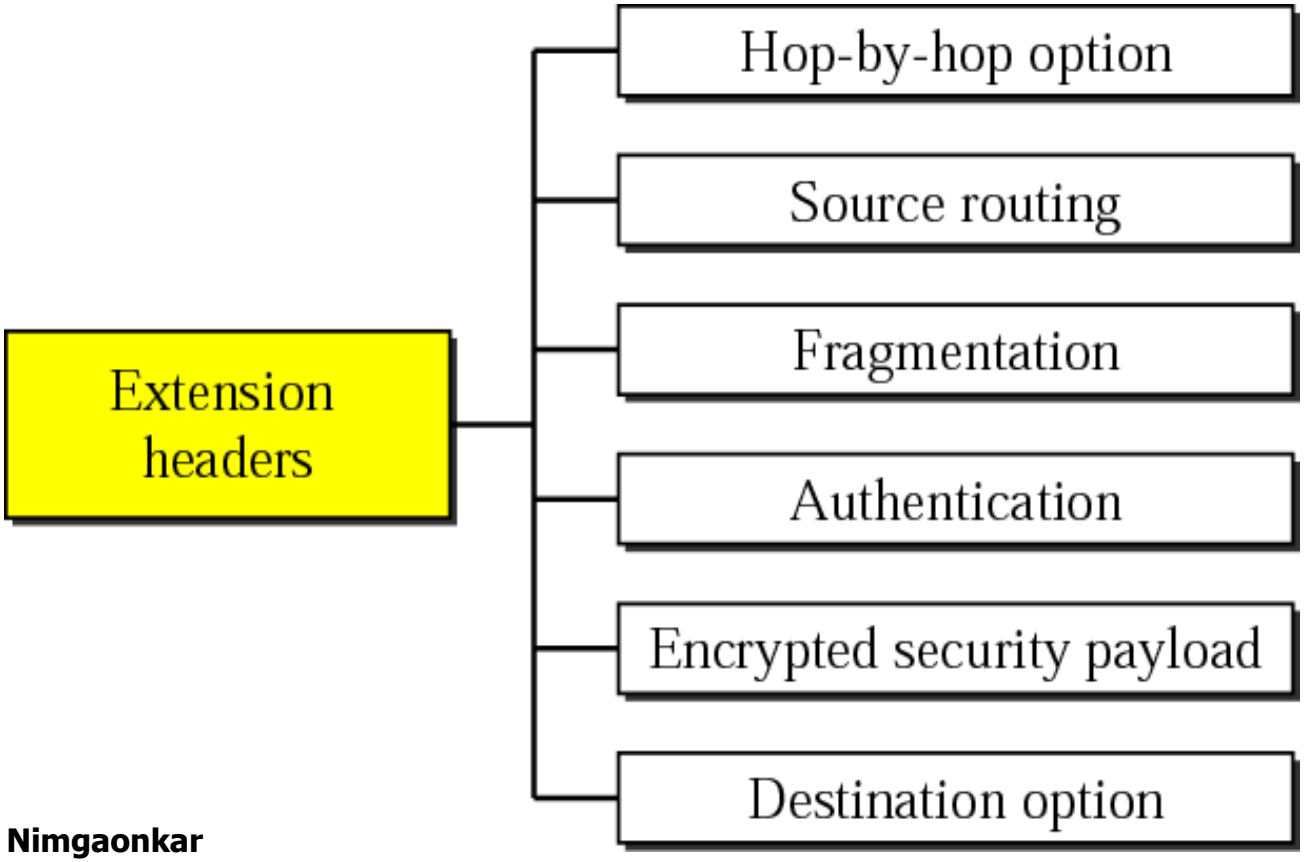
Hop-by-hop option header format

- Used when the source needs **to pass information to all routers** visited by datagram.
  - Certain management, debugging, control management
  - if length of datagram is more than 65,535 bytes



IPv6 datagram – Extension header

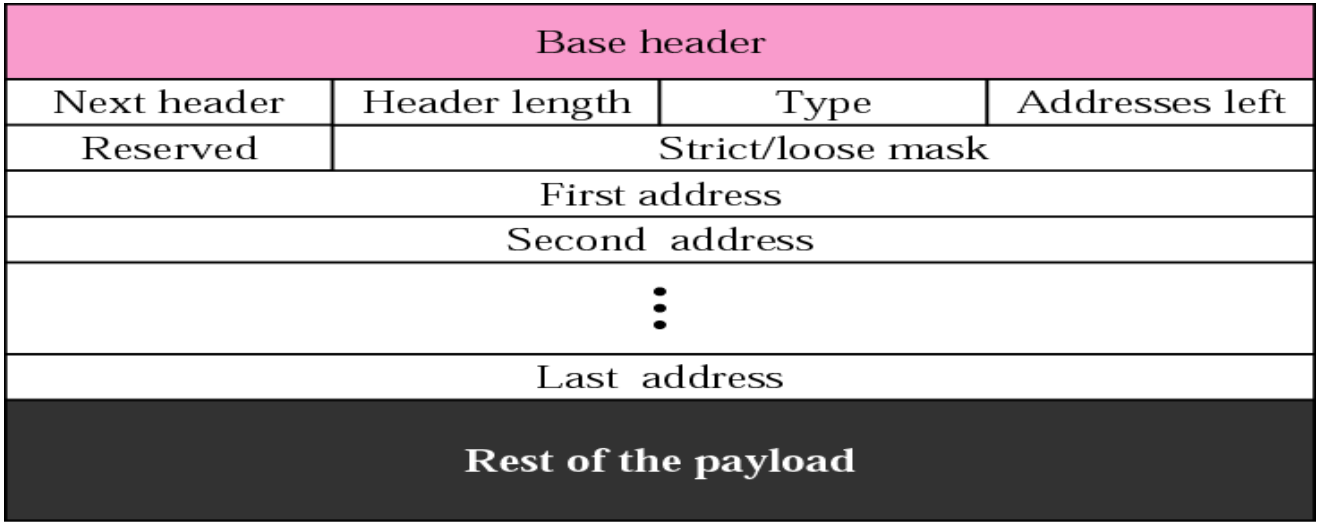
- Only present when needed.
- Eliminated IPv4's **40-byte limit on options**



Source routing

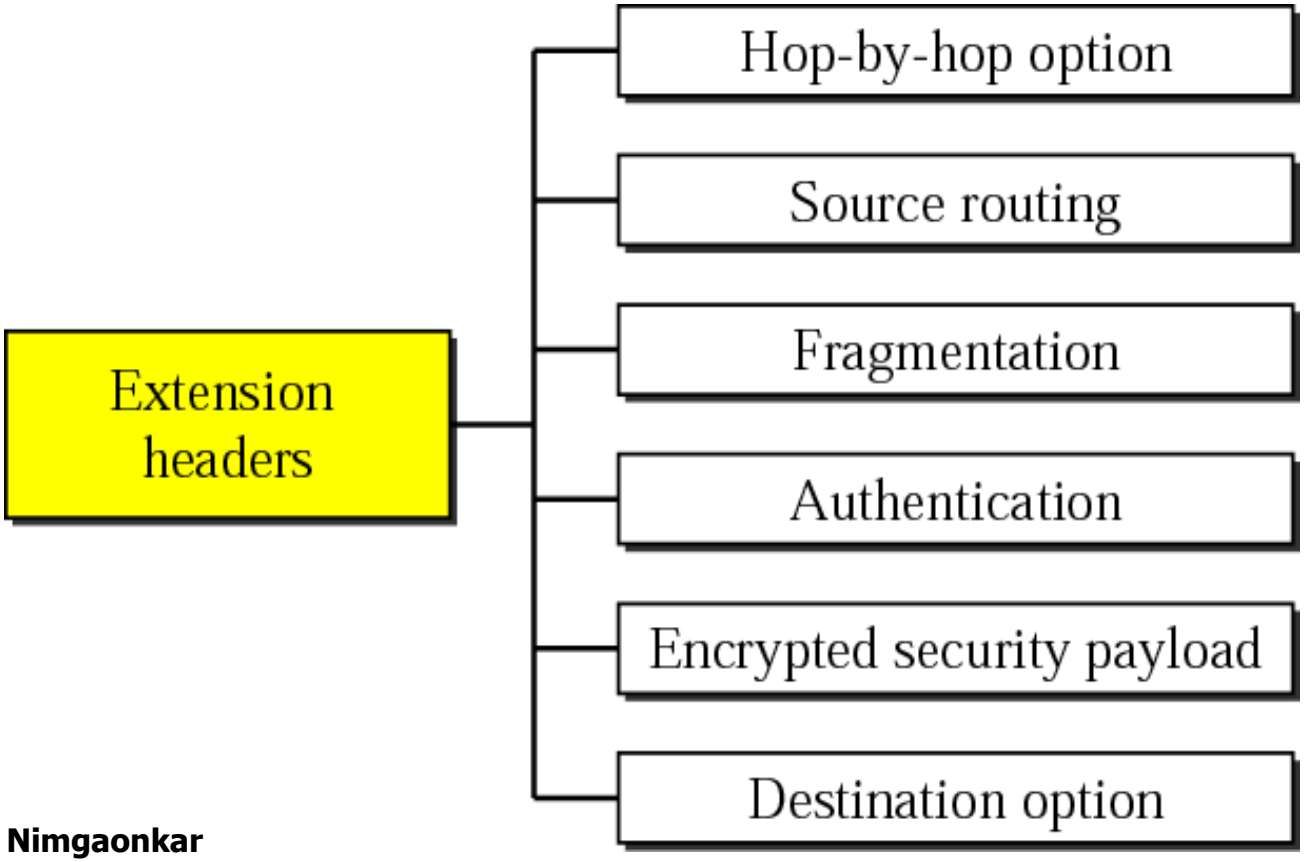
Combines **strict** and **loose** source routing of IPv4.

- **Type**: Source routing
- **Addresses left**: number of hops still remaining to reach destination.
- **Strict/ loose mask**: determines rigidity of routing.
  - Strict: routing must be exactly as informed by source
  - loose: other routers may be visited apart from mentioned in header



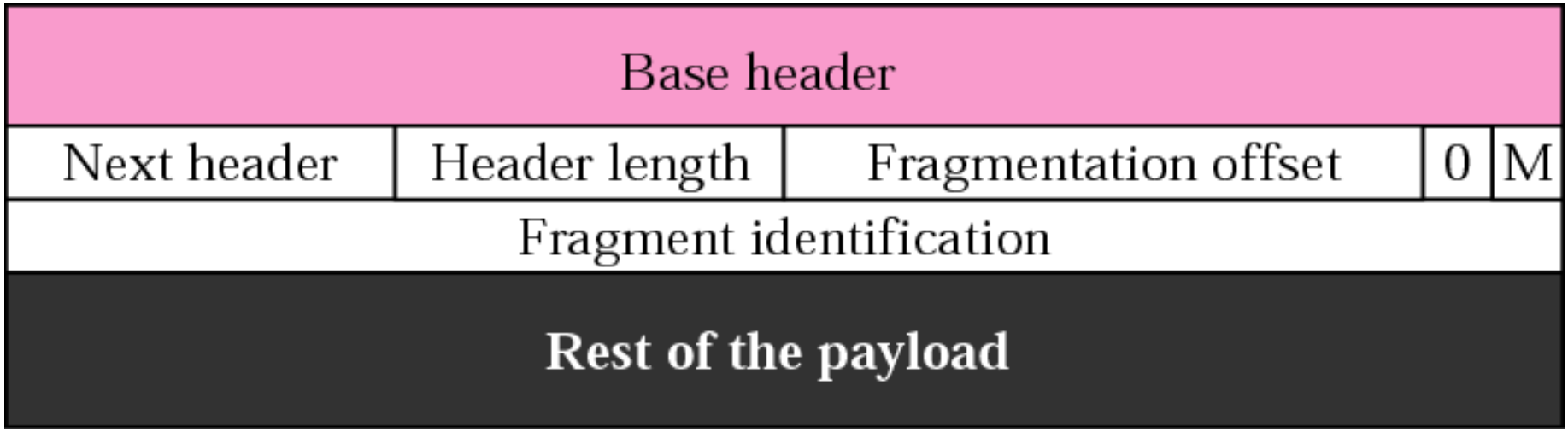
IPv6 datagram – Extension header

- Only present when needed.
- Eliminated IPv4's **40-byte limit on options**



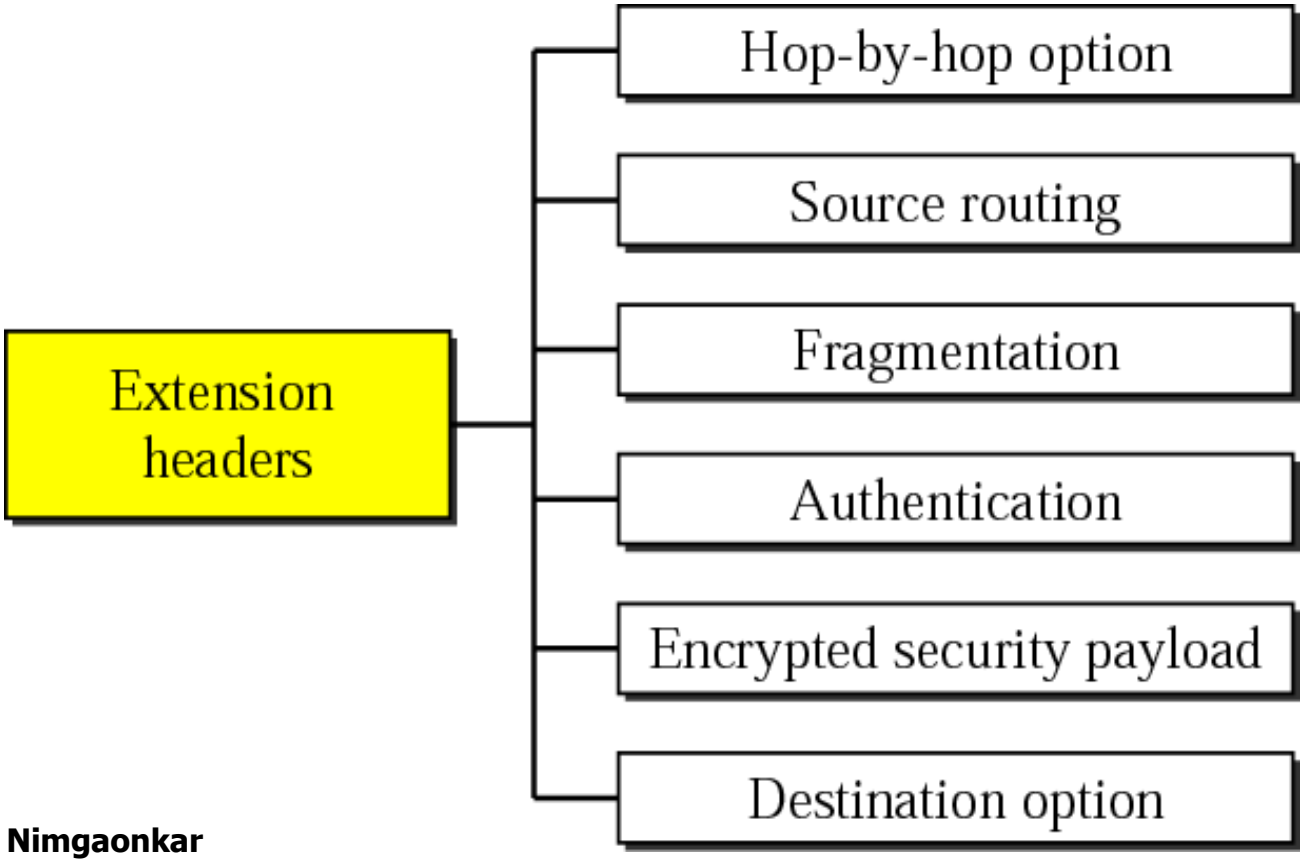
Fragmentation

- Only Original source can do the fragmentation.
- **Path MTU discovery technique** to be used to find smallest MTU on the path
- default smallest MTU size is 1280 bytes or smaller. (Size is supported by each network connected to the Internet)



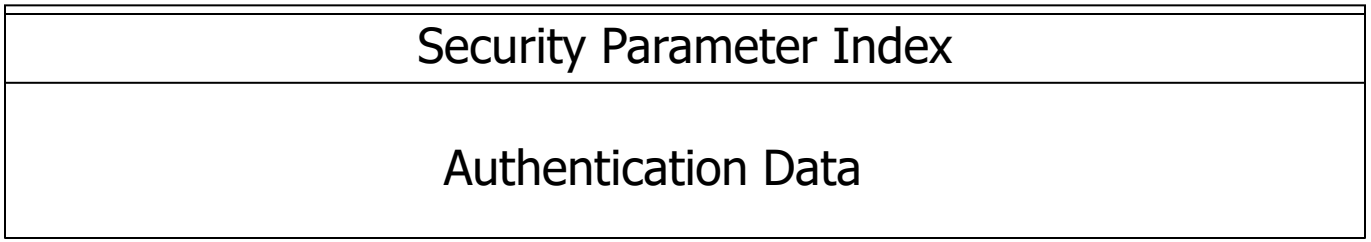
IPv6 datagram – Extension header

- Only present when needed.
- Eliminated IPv4's **40-byte limit on options**



Authentication

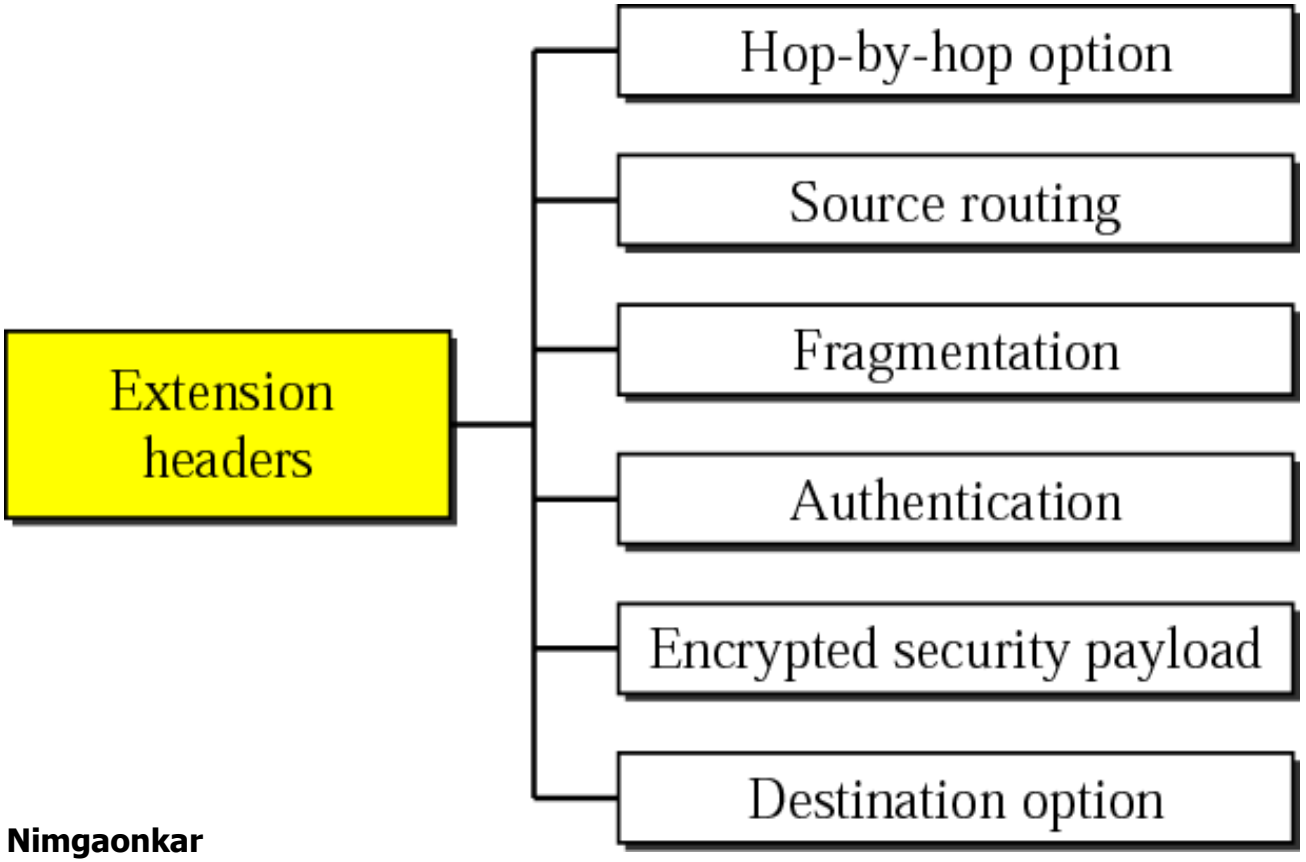
- Authentication extension header
  - **Validates the message sender**. This check ensures sender is genuine
  - **Ensures data integrity.** Ensures data is not altered by hackers.



Security Parameter Index: defines **algorithm used for authentication**
Authentication data: Actual **data generated** by algorithm.

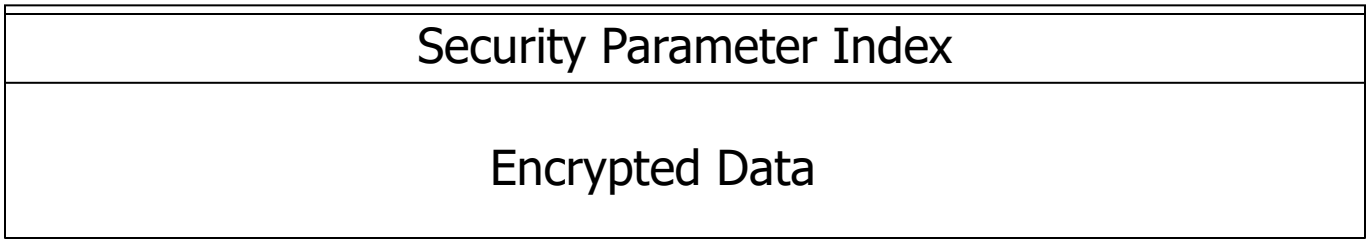
IPv6 datagram – Extension header

- Only present when needed.
- Eliminated IPv4's **40-byte limit on options**



Encrypted Security Payload

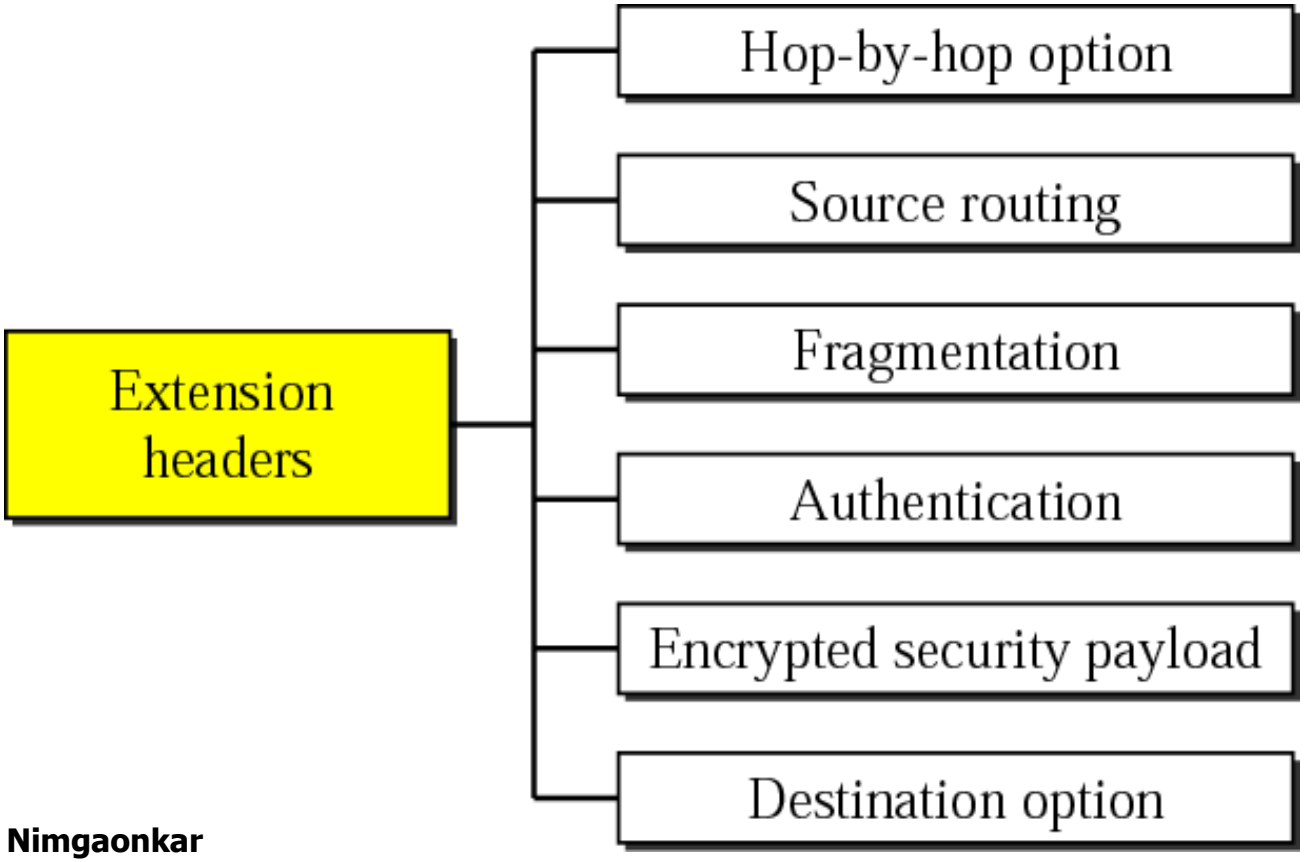
• Provides guard against confidentiality and guards against eavesdropping



Security Parameter Index: Type of Encryption and decryption used
Encrypted data: Encrypted data along with extra parameters if needed by algorithm.

IPv6 datagram – Extension header

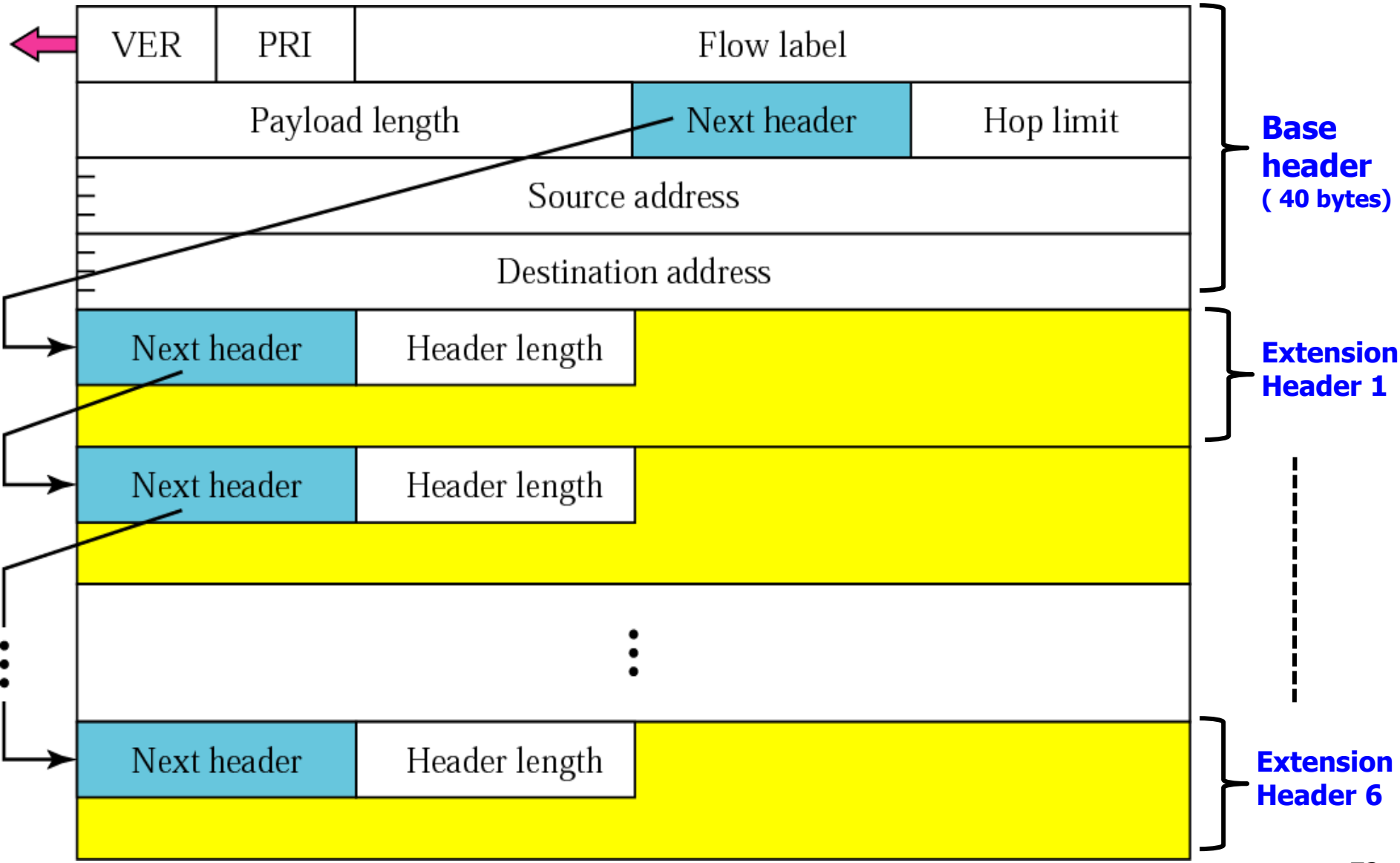
- Only present when needed.
- Eliminated IPv4's **40-byte limit on options**



Destination Option

- Used when source needs to **pass information to destination only**.
- Intermediate **routers are restricted** accessing this info.
- Format is same as hop by hop option.

IPv6 datagram – Base & Extension header



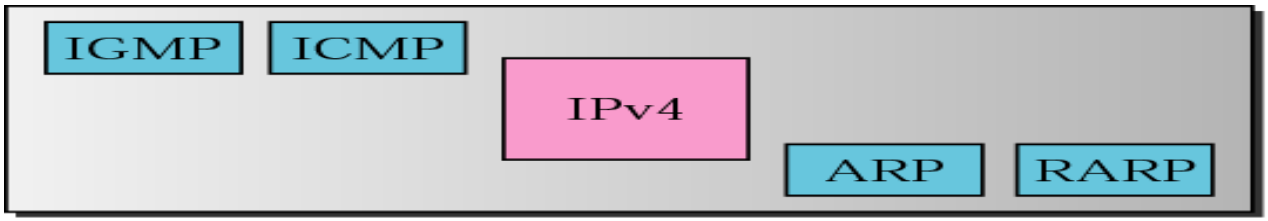
Internet Control Message Protocol Version 6

ICMPv6

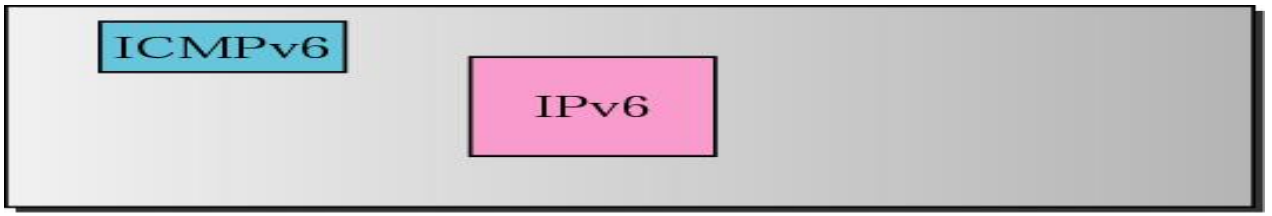
ICMPv6

ICMPv6, while similar in strategy to ICMPv4, has changes that makes it more suitable for IPv6.

ICMPv6 has absorbed some protocols that were independent in IPv4, ARP, ICMP and IGMP are merged in ICMPv6



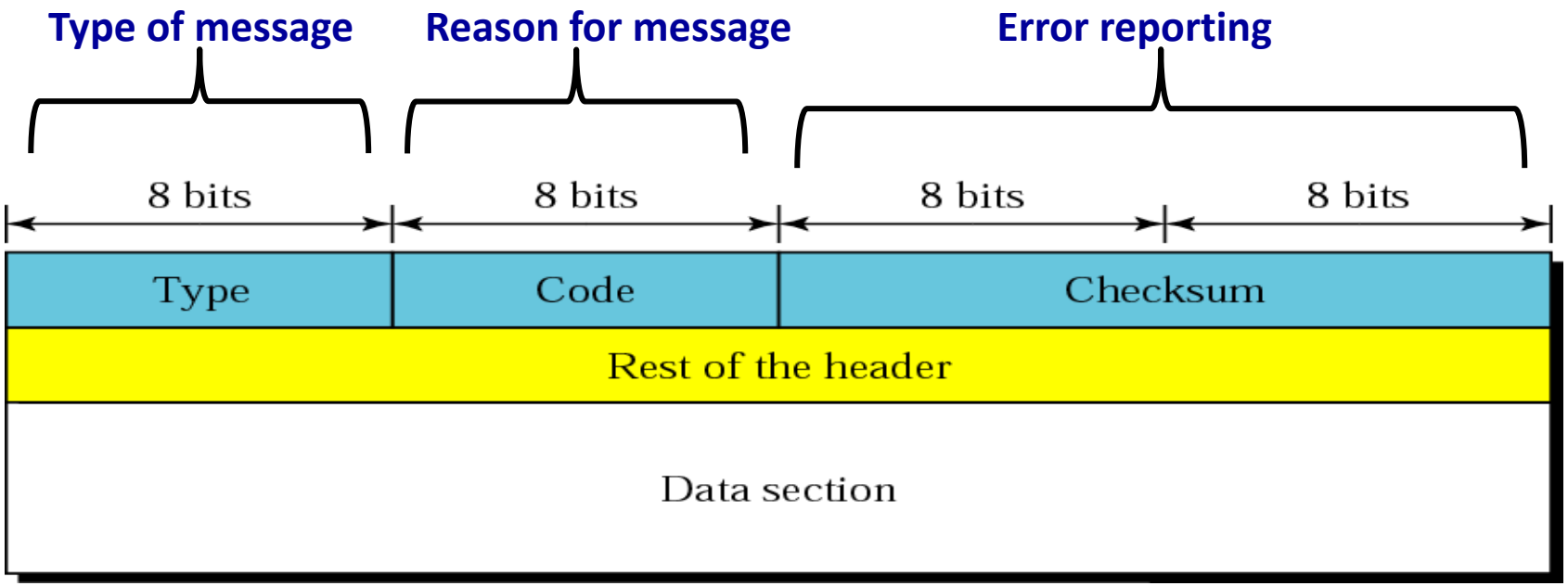
Network layer in version 4



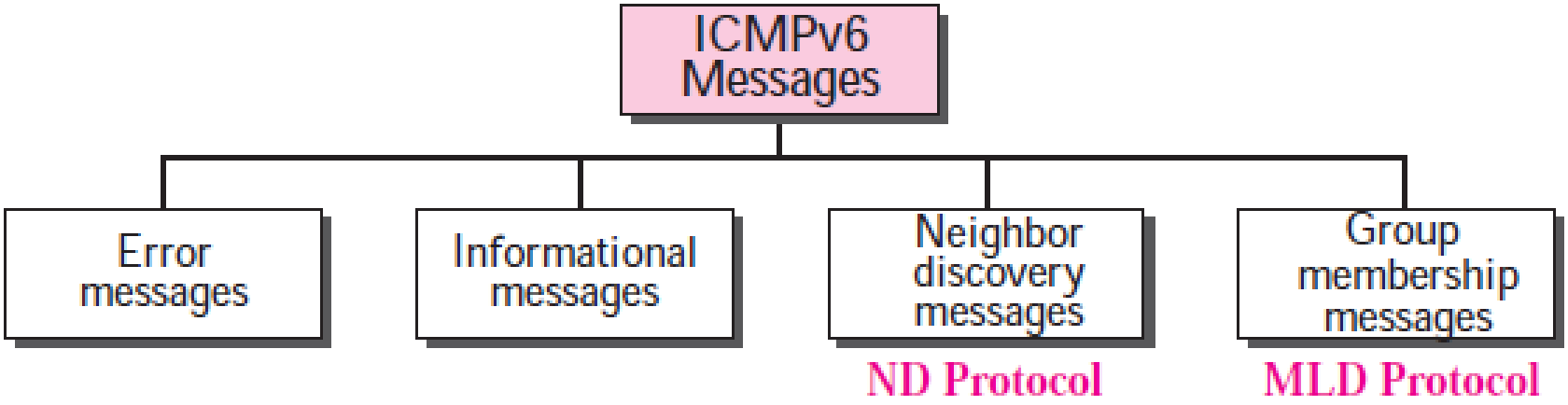
Network layer in version 6

Message Format

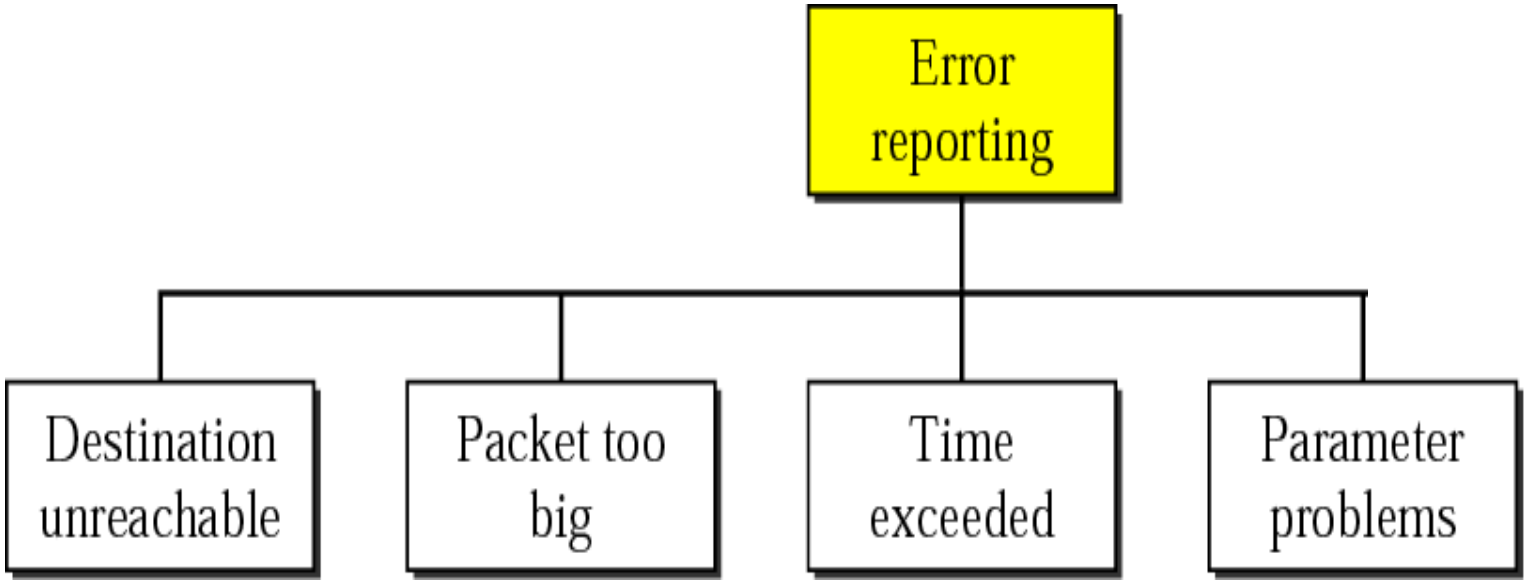
- 8-byte header and a variable-size data section.
- The general format of the header is different for each message type, the first 4 bytes are common to all.



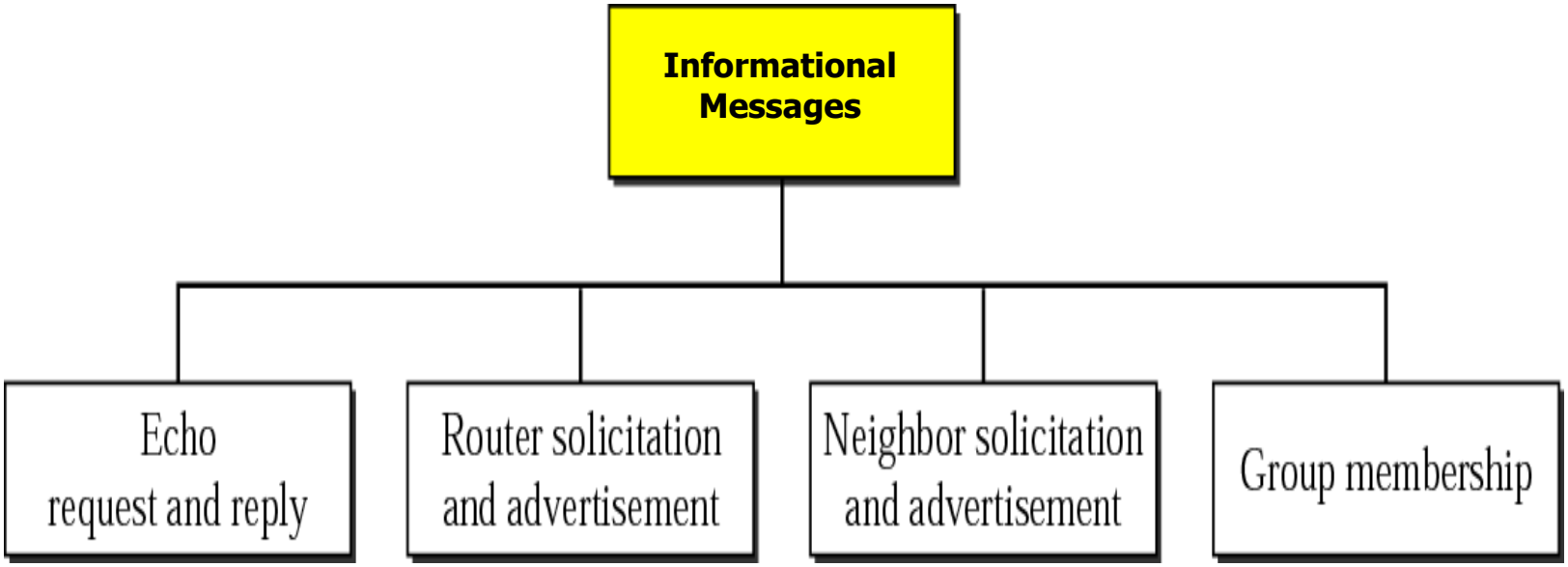
Types of ICMPv6 messages



Types of ICMP message - Error Reporting



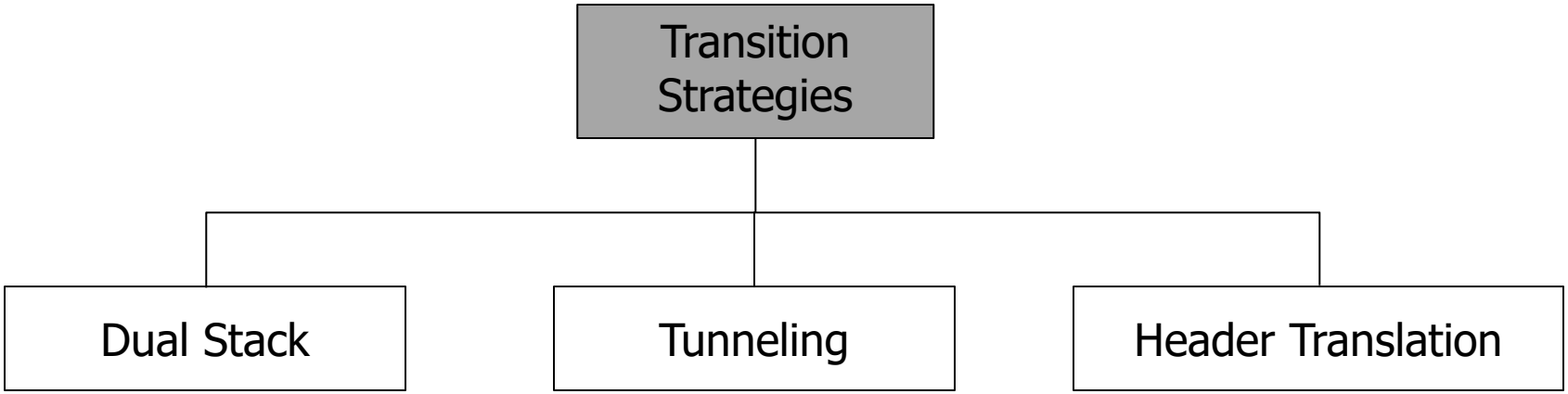
Informational Messages



Transition from IPv4 to IPv6

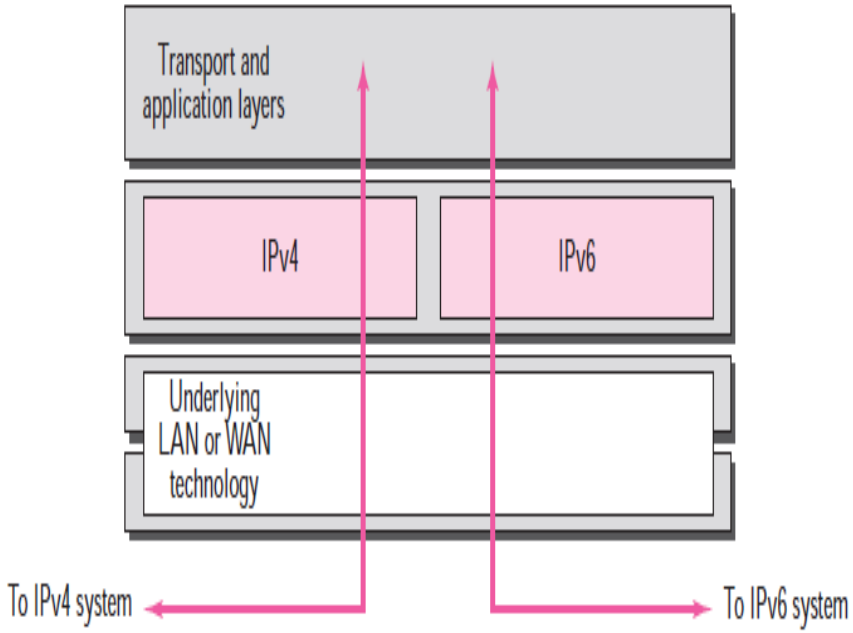
Transition from IPv4 to IPv6

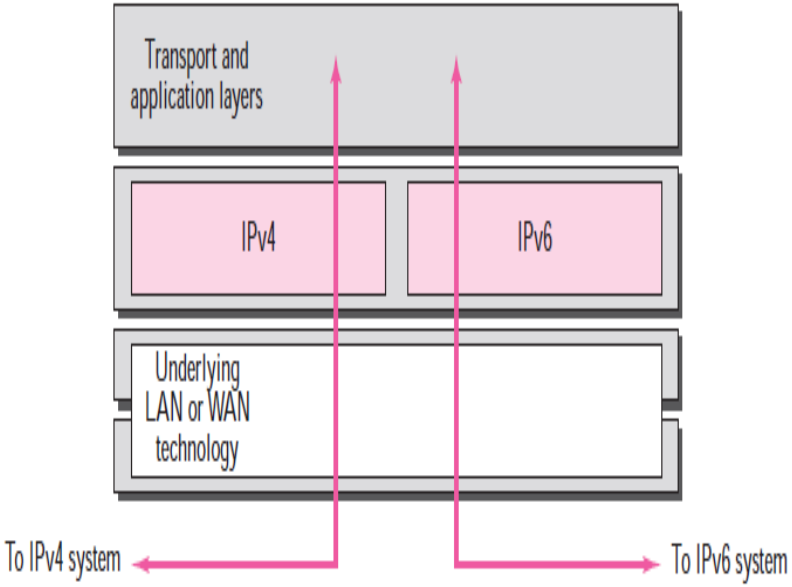
- Transition will not be easy, as there are huge number of systems on the internet.
- It will take considerable amount of time



Dual Stack

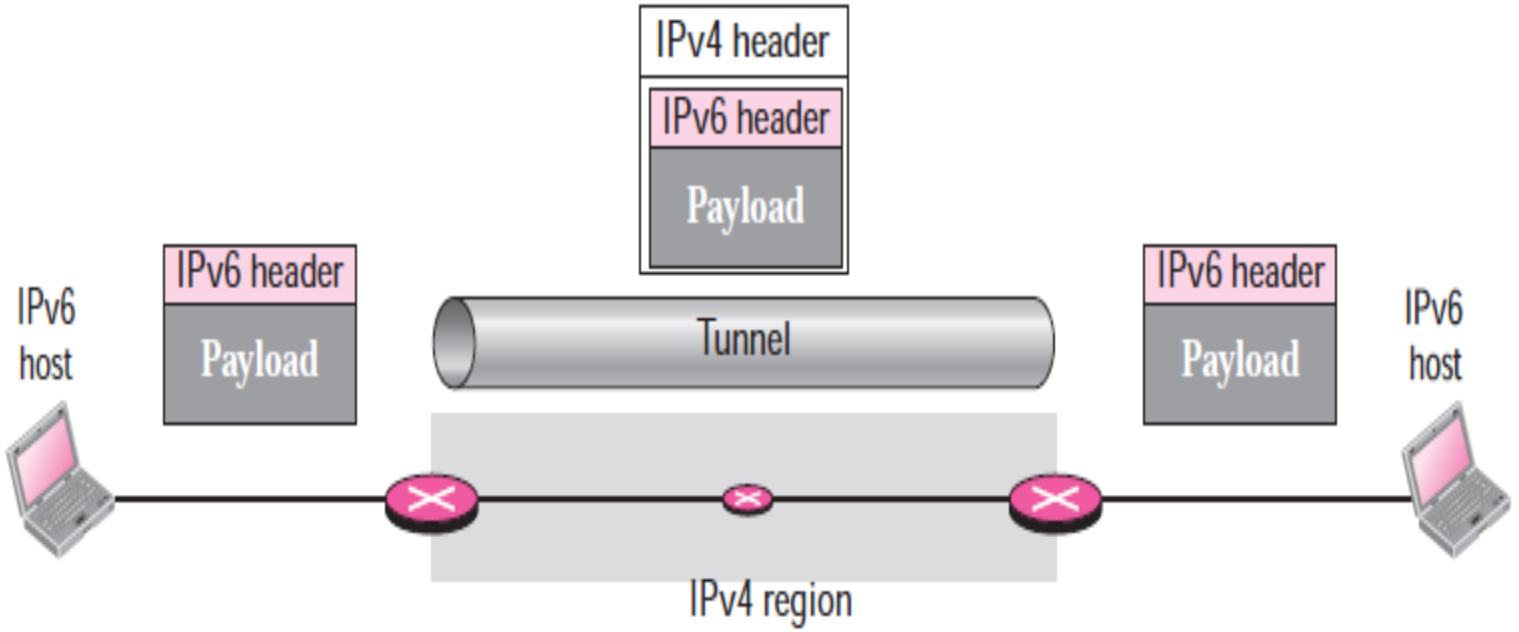
- Both IPv4 and IPv6 run on machine.
- Which version is to be used to send packet is determined by querying to **DNS**





Tunneling

- It is used when two hosts using IPv6 wants to communicate with each other and the data **packet has to pass through IPv4 network**
- In this IPv6 packet is encapsulated in IPv4



Header Translation

- Sender is IPv6
- Receiver is IPv4
- mapped header translation takes place