# COMPUTER NETWORK PROTOCOL

## Switching

### Design of Cross bar Switch

To connects n inputs to m outputs in a gridusing crossbar switch requires n × m crosspoints

### Design of Multistage Swtich

For a three stage switch with N input and N output, total number of crosspoints is

$2kN+k(N/n)^2$

To design a three-stage switch, these steps should be followed:

1. Divide the N input lines into groups, each of n lines. For each group, use one crossbar of size n × k, where k is the number of crossbars in the middle stage. In other words, the first stage has N/n crossbars of n × k crosspoints.
2. Use k crossbars, each of size (N/n) × (N/n) in the middle stage.
3. Use N/n crossbars, each of size k × n at the third stage.

### Design of Banyan Switch

For n inputs and n outputs, it has $\log_2 n$ stages with n/2 microswitches at each stage.

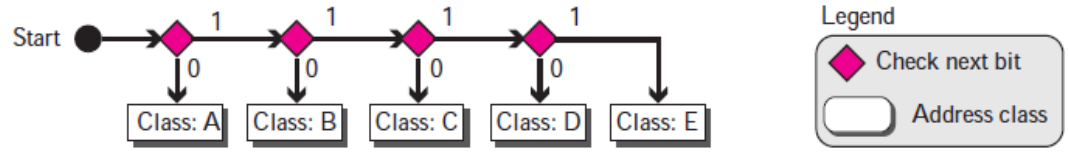
### Clos Criteria

$n = (N)^{1/2}$ and $k \geq 2n-1$

Total number of crosspoints $\geq 4N [(2N)^{1/2} -1]$

## IPv4 Addressing

### Finding the address class using continuous checking




subnet mask : $n_{sub} = n + \log_2 s$ supernet mask : $n_{super} = n - \log_2 c$

### Extracting Block Information in classless addressing

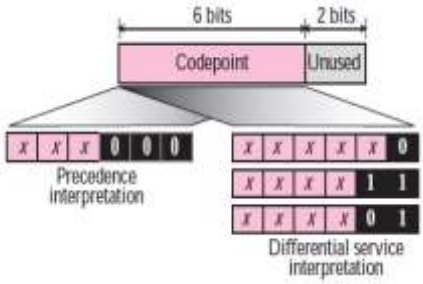
1. The number of addresses in the block, $N = 2^{32-n}$
2. First address = (any address) AND (network mask)
3. Last address = (any address) OR [NOT (network mask)]
4. Subnet mask, $n_{sub} = n + \log_2 (N/N_{sub})$

## IPv6 Addressing

### Prefixes for IPv6 Addresses

| | *Block Prefix* | *CIDR* | *Block Assignment* | *Fraction* |
|---|---|---|---|---|
| 1 | 0000 0000 | 0000::/8 | Reserved (IPv4 compatible) | 1/256 |
| | 0000 0001 | 0100::/8 | Reserved | 1/256 |
| | 0000 001 | 0200::/7 | Reserved | 1/128 |
| | 0000 01 | 0400::/6 | Reserved | 1/64 |
| | 0000 1 | 0800::/5 | Reserved | 1/32 |
| | 0001 | 1000::/4 | Reserved | 1/16 |
| **2** | **001** | **2000::/3** | **Global unicast** | **1/8** |
| 3 | 010 | 4000::/3 | Reserved | 1/8 |
| 4 | 011 | 6000::/3 | Reserved | 1/8 |
| 5 | 100 | 8000::/3 | Reserved | 1/8 |
| 6 | 101 | A000::/3 | Reserved | 1/8 |
| 7 | 110 | C000::/3 | Reserved | 1/8 |
| 8 | 1110 | E000::/4 | Reserved | 1/16 |
| | 1111 0 | F000::/5 | Reserved | 1/32 |
| | 1111 10 | F800::/6 | Reserved | 1/64 |
| | 1111 110 | FC00::/7 | Unique local unicast | 1/128 |
| | 1111 1110 0 | FE00::/9 | Reserved | 1/512 |
| | 1111 1110 10 | FE80::/10 | Link local addresses | 1/1024 |
| | 1111 1110 11 | FEC0::/10 | Reserved | 1/1024 |
| | 1111 1111 | FF00::/8 | Multicast addresses | 1/256 |

**Service Type**




| *Category* | *Codepoint* | *Assigning Authority* |
|---|---|---|
| 1 | XXXXX0 | Internet |
| 2 | XXXX11 | Local |
| 3 | XXXX01 | Temporary or experimental |

**Protocols field value**

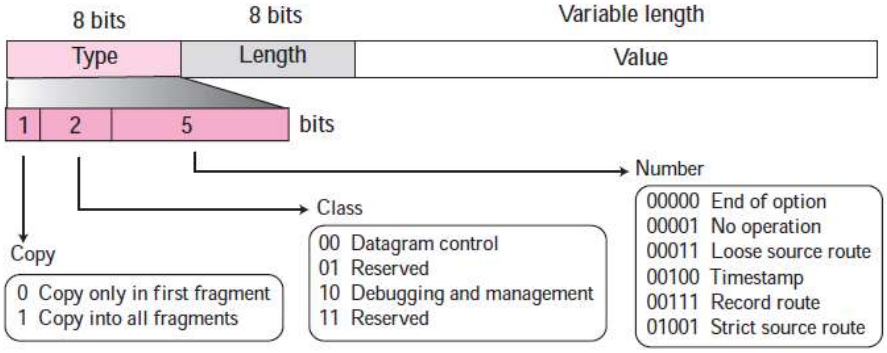
| *Value* | *Protocol* | *Value* | *Protocol* |
|---|---|---|---|
| 1 | ICMP | 17 | UDP |
| 2 | IGMP | 89 | OSPF |
| 6 | TCP | | |

**Flags field**

D : Do not fragment if value is 1.

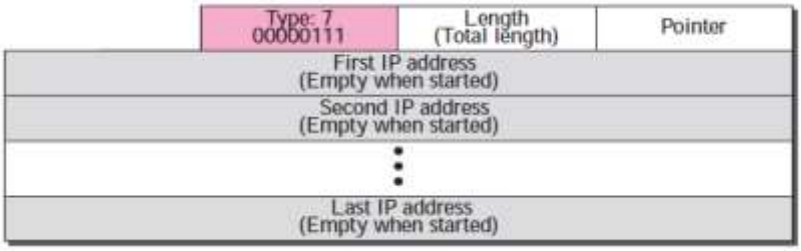
M : If value is 1 then the fragment is not the last fragment.

**Options**

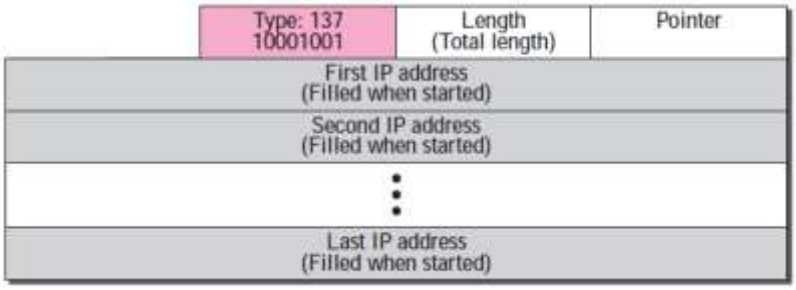



**No operation option and End-of-option option**

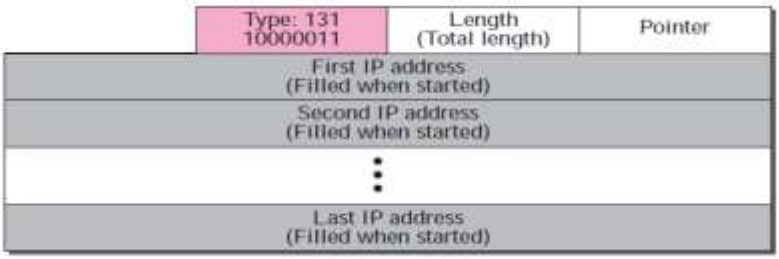

**Record-route option**



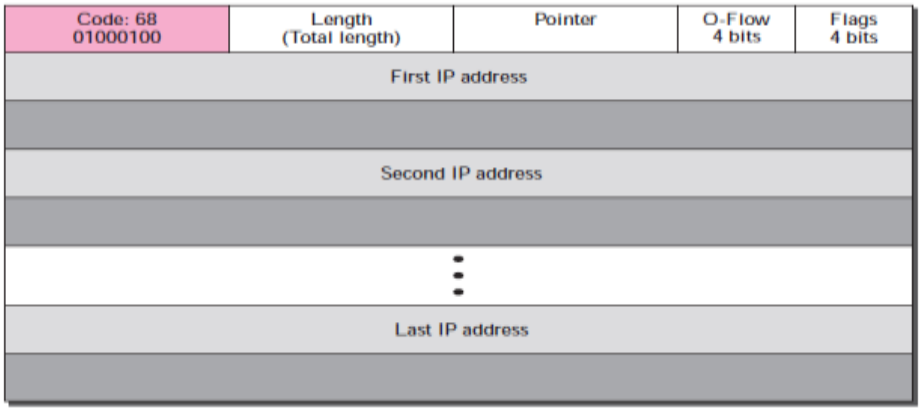
**Strict-source-route option**




**Loose-source-route option**



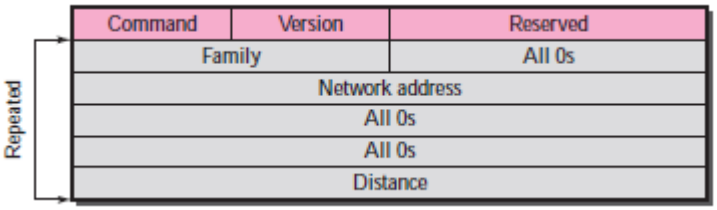

**Timestamp option**




Flag value is 0, each router adds only the timestamp in the provided field.
Flag value is 1, each router must add its outgoing IP address and the timestamp.
Flag value is 3, IP addresses given, enter timestamps

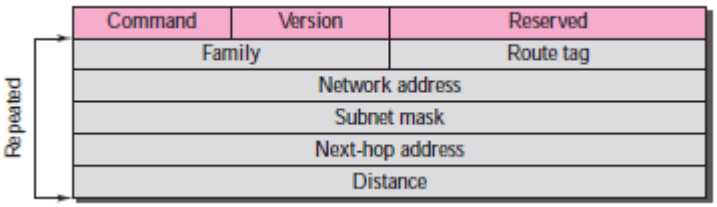
## Unicast Routing Protocols

**Routing Information Protocol (RIP) v1 Message Format:**

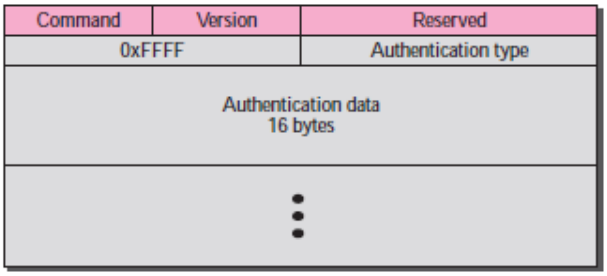



| | |
|---|---|
| Command | Request: 1, Response: 2 |
| Family | TCP/IP: 2 |

**Routing Information Protocol (RIP) v2 Message Format:**

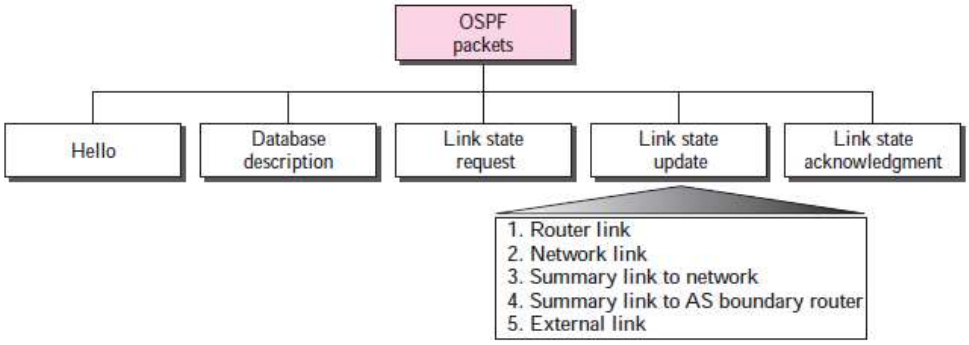



**RIPv2 Authentication**:

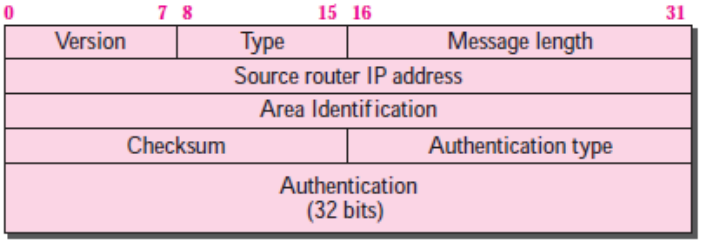



**OSPF**

## Types of OSPF Packets

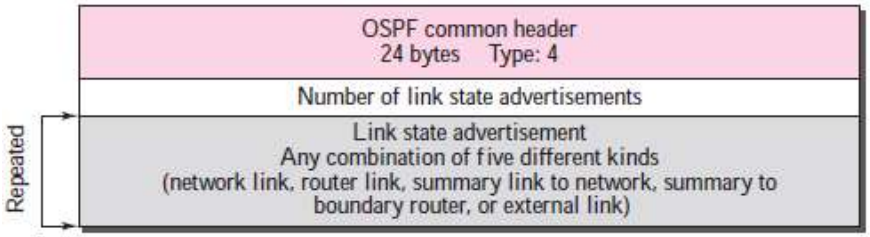



**OSPF Common Header (Version 2)**

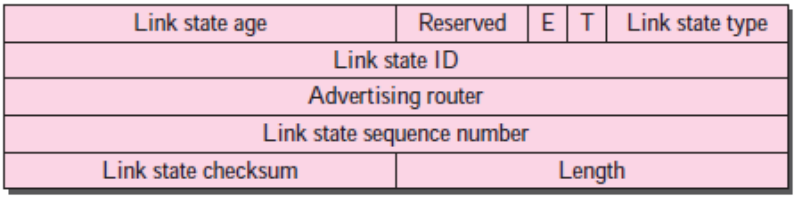



Authentication Type: 0 for None and 1 for Password.

**Link State Update Packet**

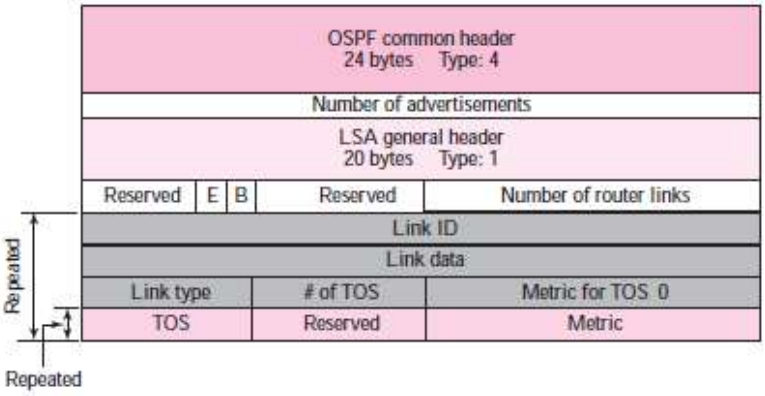



**LSA General Header**




E Flag (1 bit): 1 means Stub Area
T Flag (1 bit): 1 means router can handle multiple TOS
Link State Type: Router link (1), Network link (2), Summary link to network (3), Summary link to AS boundary router (4), and External link (5).

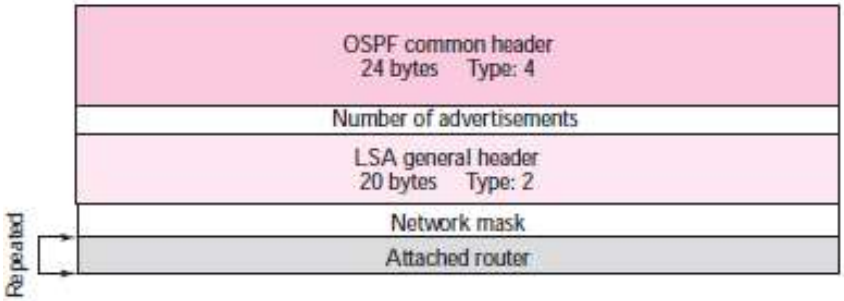
**Router Link LSA**



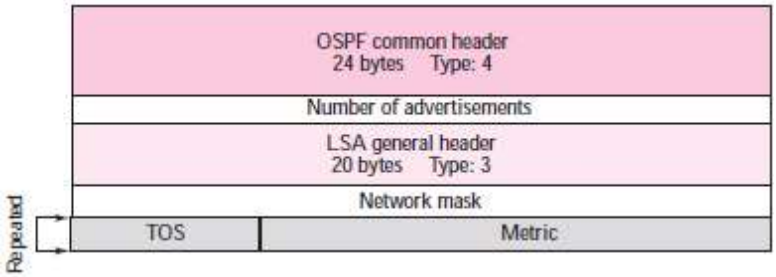

**Link ID, Link Type and Link Data**:

| *Link Type* | *Link Identification* | *Link Data* |
|---|---|---|
| Type 1: Point-to-point | Address of neighbor router | Interface number |
| Type 2: Transient | Address of designated router | Router address |
| Type 3: Stub | Network address | Network mask |
| Type 4: Virtual | Address of neighbor router | Router address |

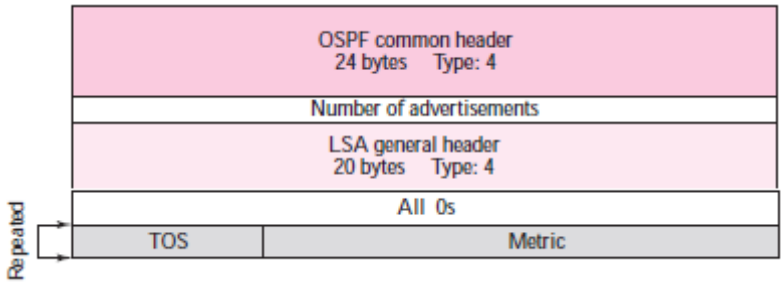
**Network Link Advertisement Format**



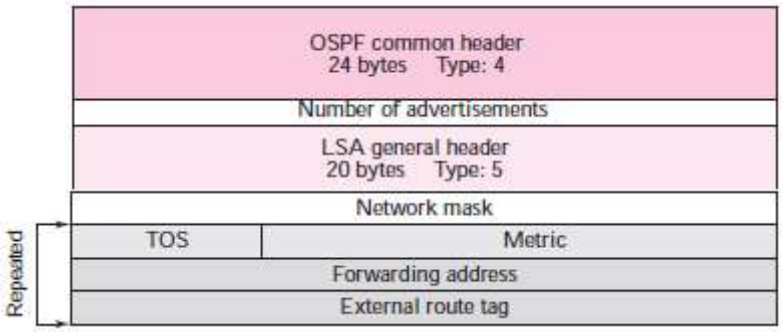
**Summary Link to Network LSA**



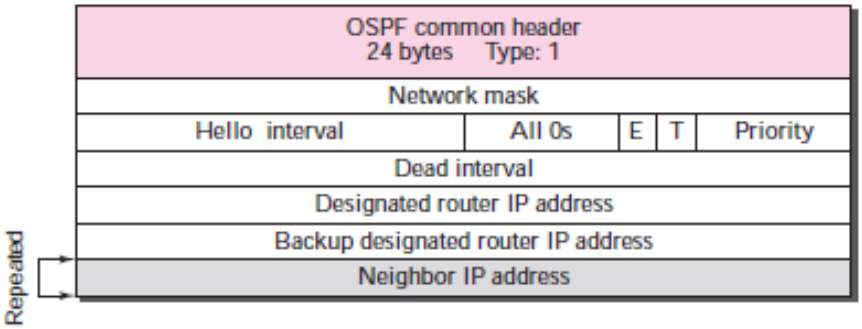

**Summary Link to AS Boundary Router LSA**




**External Link LSA**

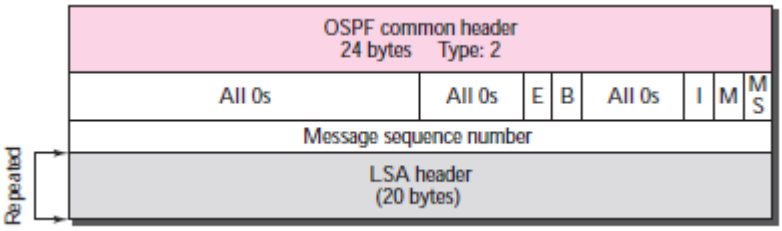



**Hello Packet**



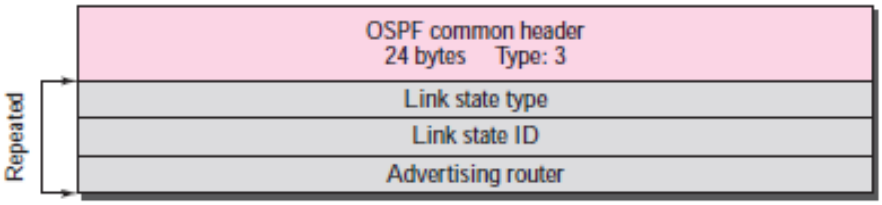

E Flag (1 bit): 1 means Stub Area
T Flag (1 bit): 1 means router can handle multiple TOS

**Database Description Packet**

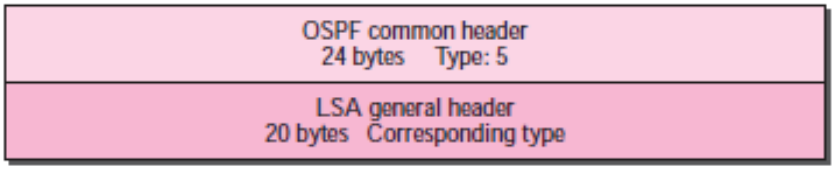



E Flag: 1 if the advertising router is an autonomous boundary router.
B Flag: 1 if the advertising router is an area border router.
I Flag: 1 if the message is the first message.
M Flag: 1 if this is not the last message.
M/S Flag: Master =1, Slave = 0

**Link State Request Packet**

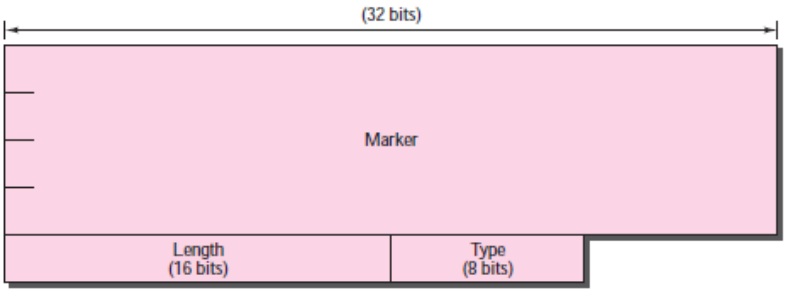



**Link State Acknowledgement Packet**

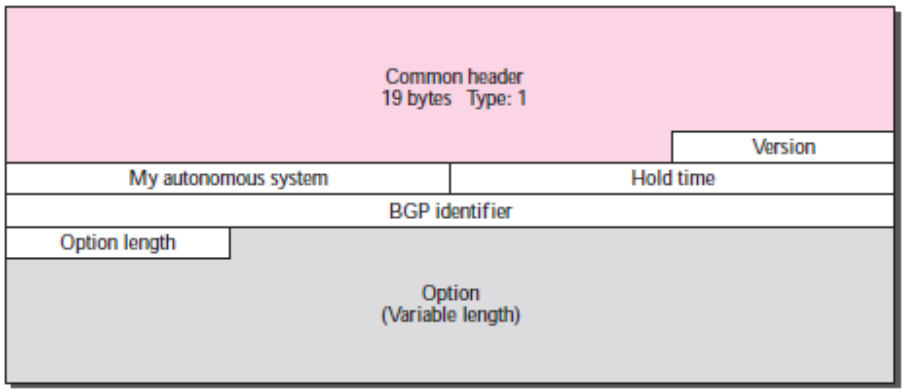



## BGP

**BGP Packet Header**

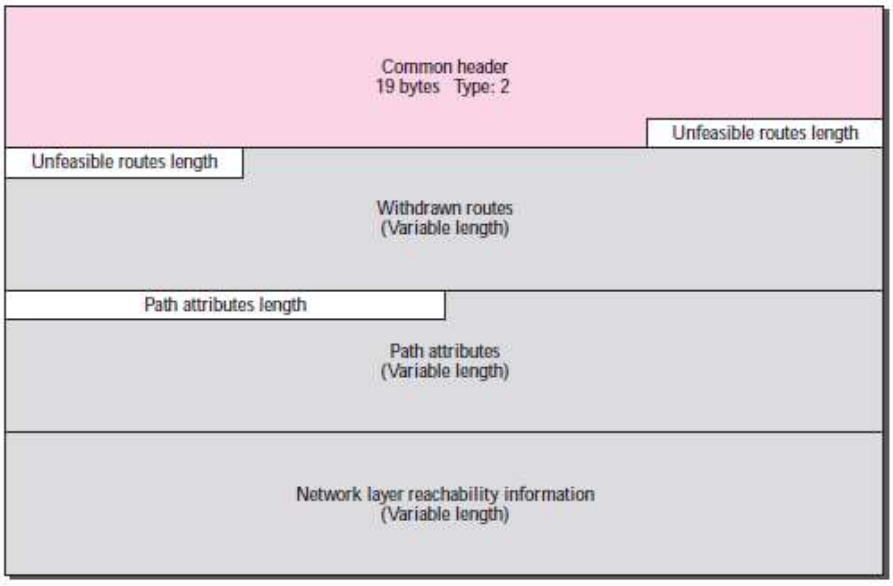


**Open Message**

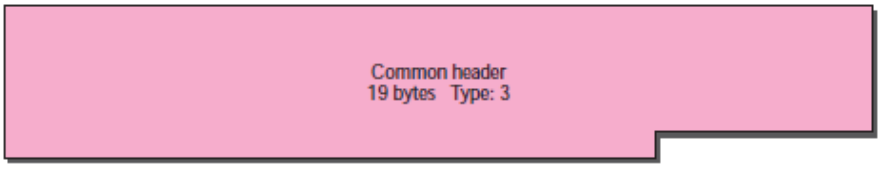



Version = 4

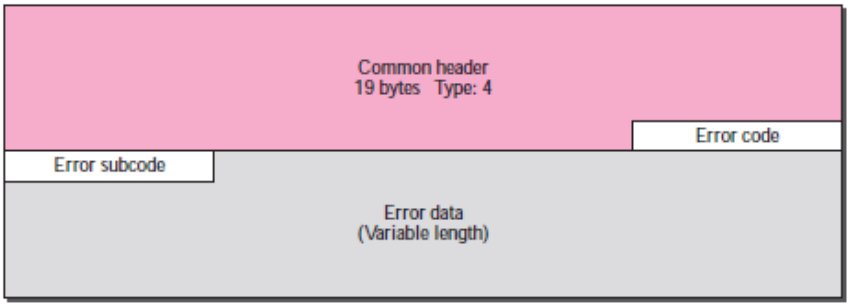
**Update Message**




**Keepalive Message**




**Notification Message**

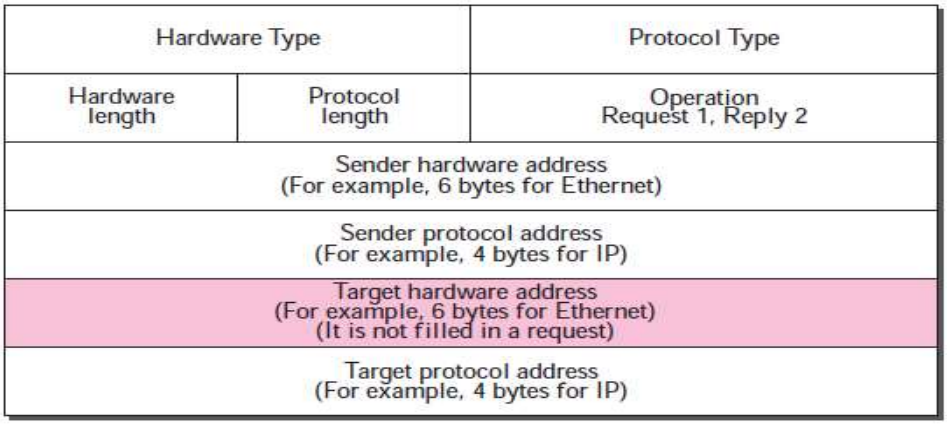


**Error Codes**

| *Error Code* | *Error Code Description* | *Error Subcode Description* |
|---|---|---|
| 1 | Message header error | Three different subcodes are defined for this type of error: synchronization problem (1), bad message length (2), and bad message type (3). |
| 2 | Open message error | Six different subcodes are defined for this type of error: unsupported version number (1), bad peer AS (2), bad BGP identifier (3), unsupported optional parameter (4), authentication failure (5), and unacceptable hold time (6). |
| 3 | Update message error | Eleven different subcodes are defined for this type of error: malformed attribute list (1), unrecognized well-known attribute (2), missing well-known attribute (3), attribute flag error (4), attribute length error (5), invalid origin attribute (6), AS routing loop (7), invalid next hop attribute (8), optional attribute error (9), invalid network field (10), malformed AS_PATH (11). |
| 4 | Hold timer expired | No subcode defined. |
| 5 | Finite state machine error | This defines the procedural error. No subcode defined. |
| 6 | Cease | No subcode defined. |

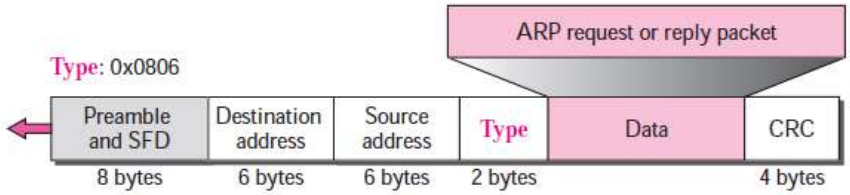
## ARP

**ARP packet**




Hardware Type :16 bits ; Protocol Type :16 bits ; Hardware length :8 bits ; Protocol length : 8 bits; Operation : 16 bits

**Encapsulation of ARP packet**

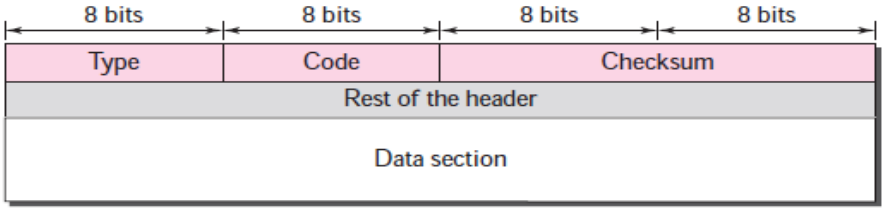


## ICMP

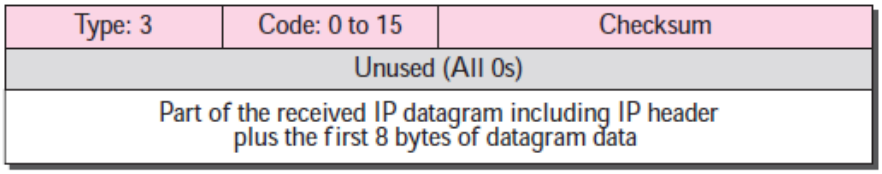
### ICMP Messages

| *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 |

### General Message format of ICMP message

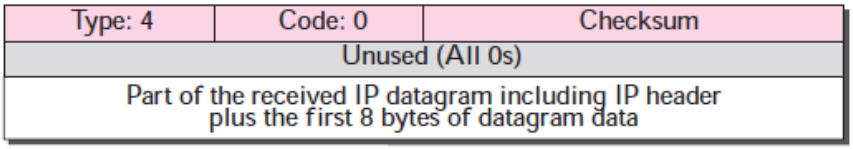



### Destination Unreachable message format

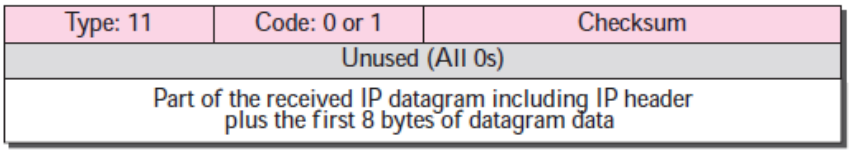



Code 0. The network is unreachable, possibly due to hardware failure.
Code 1. The host is unreachable. This can also be due to hardware failure.
Code 2. The protocol is unreachable.
Code 3. The port is unreachable.
Code 4. Fragmentation is required, but the DF (do not fragment) field of the datagram has been set.
Code 5. Source routing cannot be accomplished.
Code 6. The destination network is unknown.
Code 7. The destination host is unknown.
Code 8. The source host is isolated.
Code 9. Communication with the destination network is administratively prohibited.
Code 10. Communication with the destination host is administratively prohibited.
Code 11. The network is unreachable for the specified type of service.
Code 12. The host is unreachable for the specified type of service.
Code 13. The host is unreachable because the administrator has put a filter on it.
Code 14. The host is unreachable because the host precedence is violated.
Code 15. The host is unreachable because its precedence was cut off.

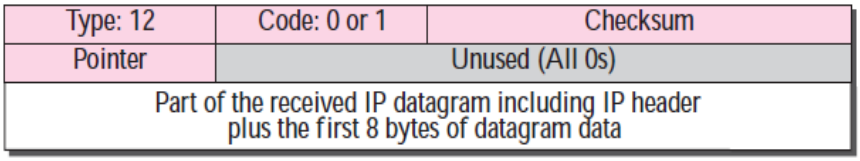
**Source Quench format**




**Time Exceeded message format**

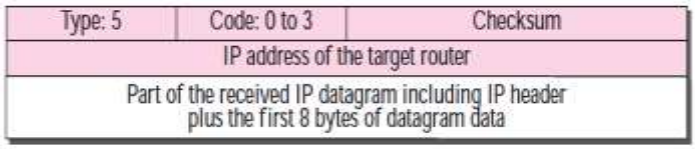



**Parameter-problem message format**




Code 0. There is an error or ambiguity in one of the header fields
Code 1. The required part of an option is missing.

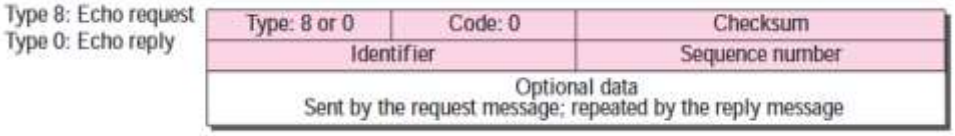
**Redirection message format**



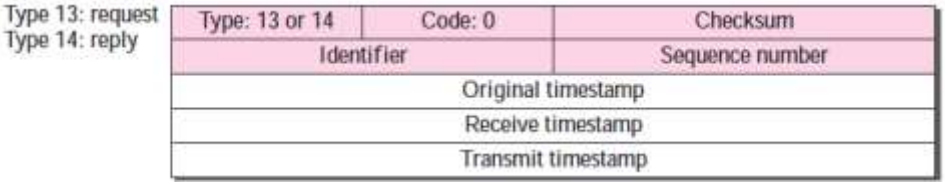

Code 0. Redirection for a network-specific route.
Code 1. Redirection for a host-specific route.
Code 2. Redirection for a network-specific route based on a specified type of service.
Code 3. Redirection for a host-specific route based on a specified type of service.

## Query Messages

**Echo request – reply messages**



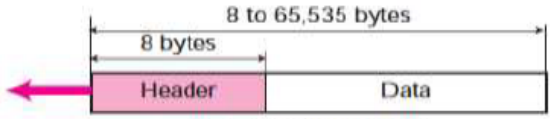

**Timestamp request reply messages**



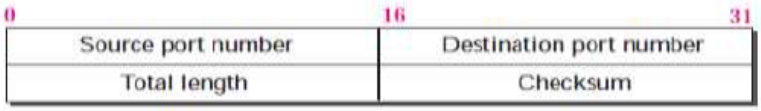
Calculations :
Sending time = receive timestamp – original timestamp
Receiving time = returned time – transmit timestamp
Round-trip time = sending time + receiving time
Time difference = receive timestamp – (original timestamp field + one way time duration)

## UDP

### User Datagram format



a. UDP user datagram



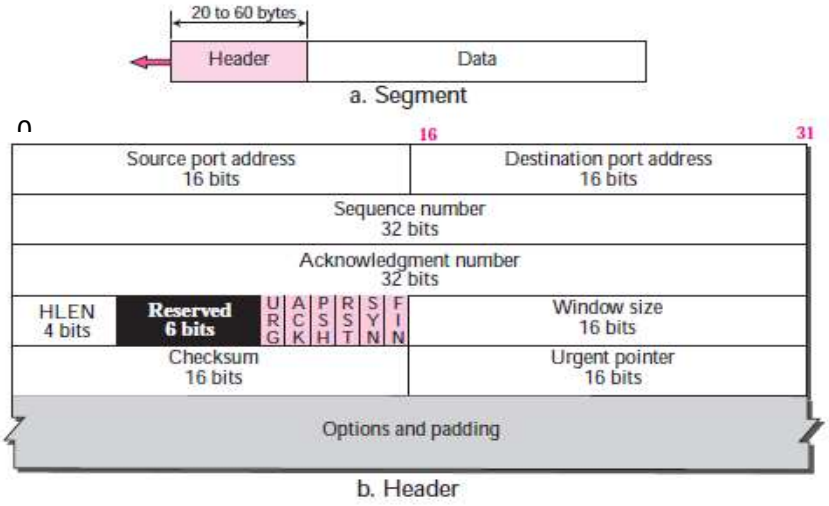
b. Header format

## TCP

### Well Known Ports

| *Port* | *Protocol* | *Description* |
|---|---|---|
| 7 | Echo | Echoes a received datagram back to the sender |
| 9 | Discard | Discards any datagram that is received |
| 11 | Users | Active users |
| 13 | Daytime | Returns the date and the time |
| 17 | Quote | Returns a quote of the day |
| 19 | Chargen | Returns a string of characters |
| 20 and 21 | FTP | File Transfer Protocol (Data and Control) |
| 23 | TELNET | Terminal Network |
| 25 | SMTP | Simple Mail Transfer Protocol |
| 53 | DNS | Domain Name Server |
| 67 | BOOTP | Bootstrap Protocol |
| 79 | Finger | Finger |
| 80 | HTTP | Hypertext Transfer Protocol |

**Segment Structure**




**RTT**

**Smoothed RTT, $RTT_S$:**
After first measurement → $RTT_S = RTT_M$ where $RTT_M$ means Measured RTT
After each measurement → $RTT_S = (1-\alpha)\ RTT_S + \alpha \times RTT_M$
Generally, $\alpha = 1/8$

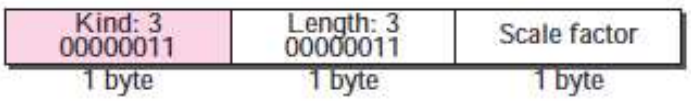
**RTT Deviation, $RTT_D$:**
After first measurement → $RTT_D = RTT_M/2$
After each measurement → $RTT_S = (1-\beta)\ RTT_D + \beta \times |RTT_S - RTT_M|$
Generally, $\beta = 1/4$

**Retransmission Time-out (RTO):**
After any measurement → $RTO = RTT_S + 4 \times RTT$

**Options**

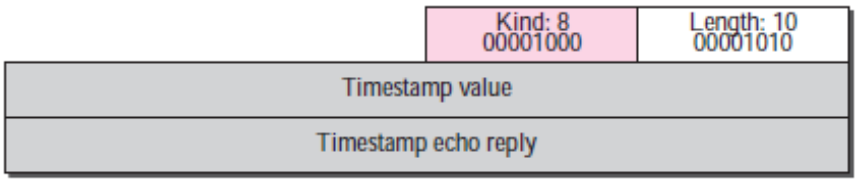
1. End-of-option: All zeros
2. No-operation option: Last bit is 1.
3. Maximum-segment-size option


4. Window-scale-factor option

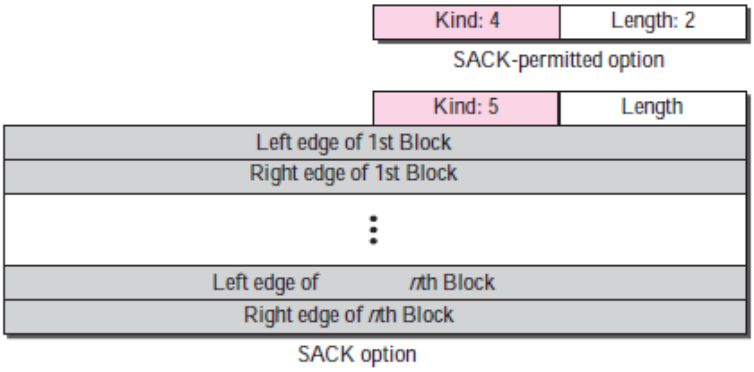



New window size = window size defined in header x $2^{\text{window scale factor}}$

5. Timestamp option

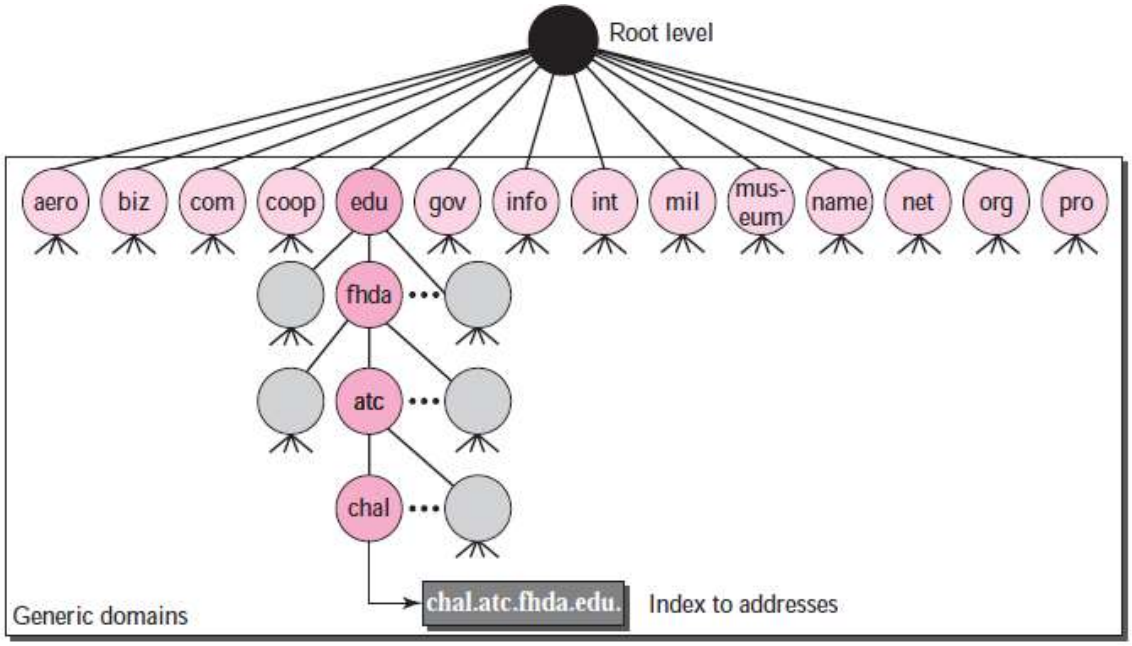


6.SACK



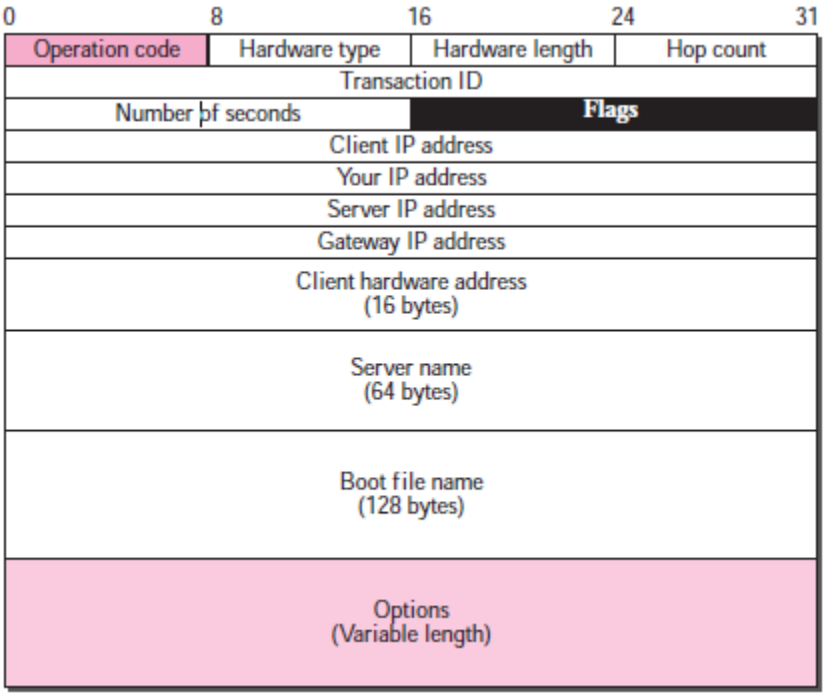

**DNS**

**Generic domains**

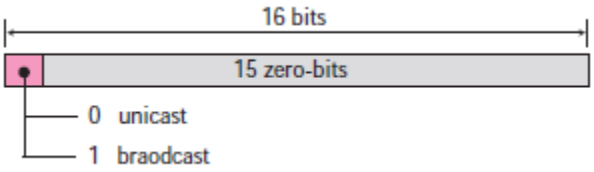


## DHCP

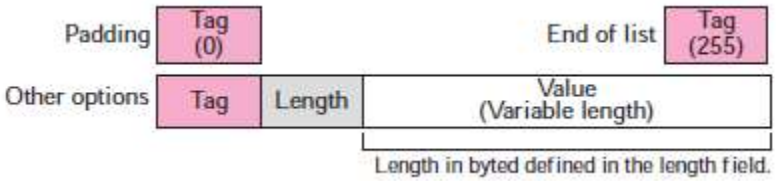
### DHCP Packet Format




### Flag Format



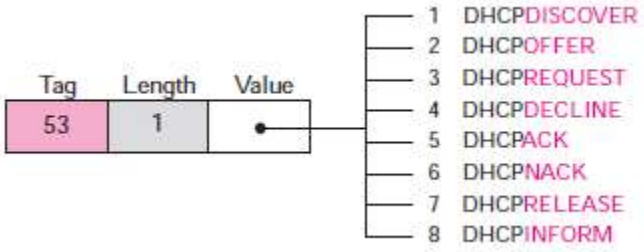

### Option Format



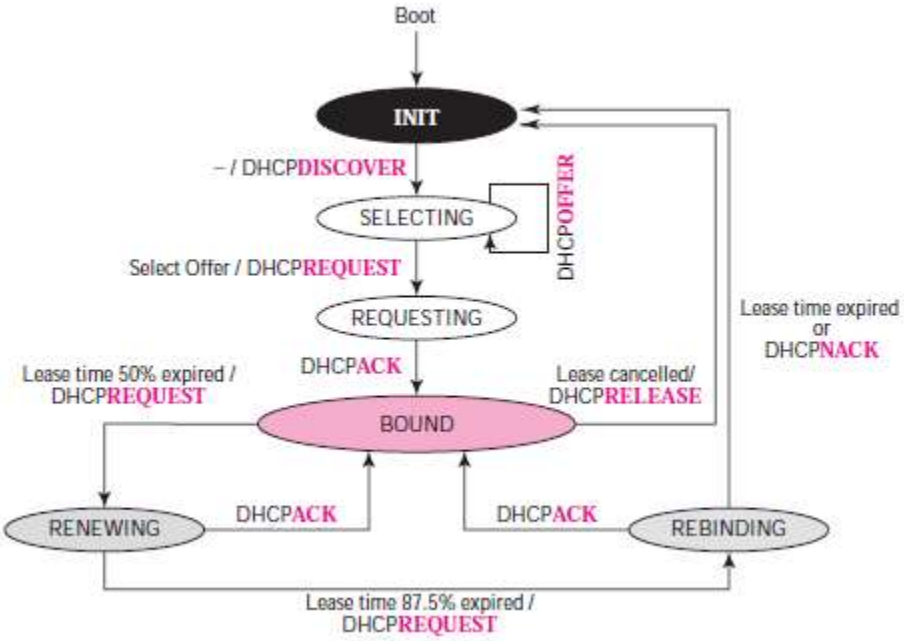
**Tables for List of Options**

| *Tag* | *Length* | *Value* | *Description* |
|---|---|---|---|
| 0 | | | Padding |
| 1 | 4 | Subnet mask | Subnet mask |
| 2 | 4 | Time of the day | Time offset |
| 3 | Variable | IP addresses | Default router |
| 4 | Variable | IP addresses | Time server |
| 5 | Variable | IP addresses | IEN 16 server |
| 6 | Variable | IP addresses | DNS server |
| 7 | Variable | IP addresses | Log server |
| 8 | Variable | IP addresses | Quote server |
| 9 | Variable | IP addresses | Print server |
| 10 | Variable | IP addresses | Impress |
| 11 | Variable | IP addresses | RLP server |
| 12 | Variable | DNS name | Host name |
| 13 | 2 | Integer | Boot file size |
| 53 | 1 | Discussed later | Used for dynamic configuration |
| 128–254 | Variable | Specific information | Vendor specific |
| 255 | | | End of list |

**Options with tag 53**



**DHCP Client Transition Diagram**




## TELNET

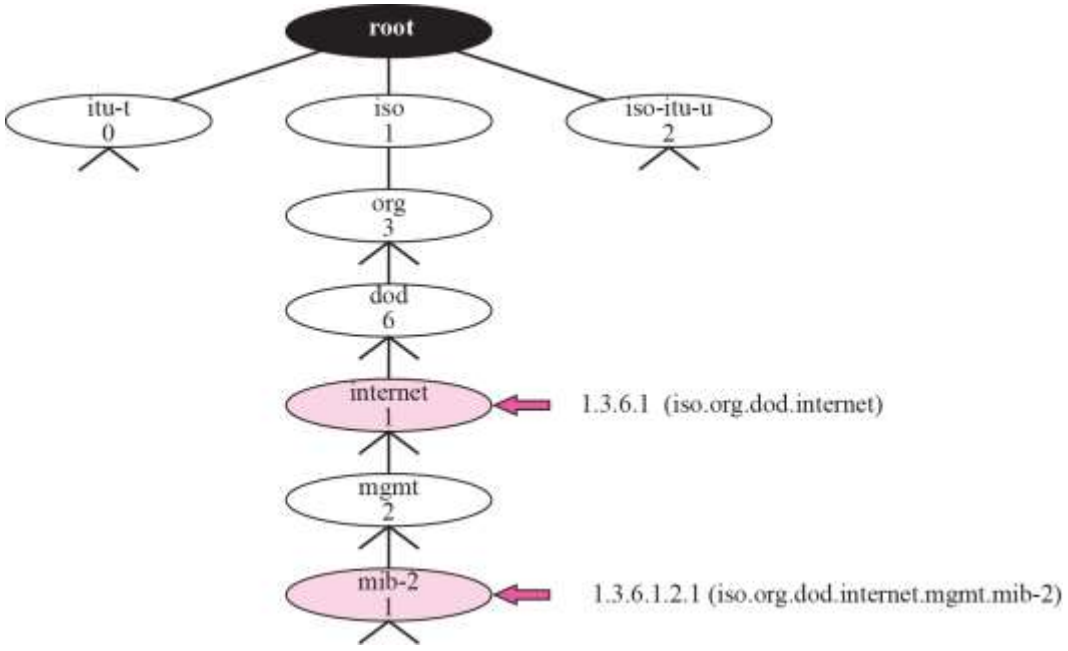
**NVT control characters**

| *Character* | *Decimal* | *Binary* | *Meaning* |
|---|---|---|---|
| EOF | 236 | 11101100 | End of file |
| EOR | 239 | 11101111 | End of record |
| SE | 240 | 11110000 | Suboption end |
| NOP | 241 | 11110001 | No operation |
| DM | 242 | 11110010 | Data mark |
| BRK | 243 | 11110011 | Break |
| IP | 244 | 11110100 | Interrupt process |
| AO | 245 | 11110101 | Abort output |
| AYT | 246 | 11110110 | Are you there? |
| EC | 247 | 11110111 | Erase character |
| EL | 248 | 11111000 | Erase line |
| GA | 249 | 11111001 | Go ahead |
| SB | 250 | 11111010 | Suboption begin |
| WILL | 251 | 11111011 | Agreement to enable option |
| WONT | 252 | 11111100 | Refusal to enable option |
| DO | 253 | 11111101 | Approval to option request |
| DONT | 254 | 11111110 | Denial of option request |
| IAC | 255 | 11111111 | Interpret (the next character) as control |

**SNMP data type**

| *Type* | *Size* | *Description* |
|---|---|---|
| INTEGER | 4 bytes | An integer with a value between $-2^{31}$ and $2^{31}-1$ |
| Integer32 | 4 bytes | Same as INTEGER |
| Unsigned32 | 4 bytes | Unsigned with a value between 0 and $2^{32}-1$ |
| OCTET STRING | Variable | Byte-string up to 65,535 bytes long |
| OBJECT IDENTIFIER | Variable | An object identifier |
| IPAddress | 4 bytes | An IP address made of four integers |
| Counter32 | 4 bytes | An integer whose value can be incremented from zero to $2^{32}$; when it reaches its maximum value it wraps back to zero |
| Counter64 | 8 bytes | 64-bit counter |
| Gauge32 | 4 bytes | Same as Counter32, but when it reaches its maximum value, it does not wrap; it remains there until it is reset |
| TimeTicks | 4 bytes | A counting value that records time in 1/100ths of a second |
| BITS | | A string of bits |
| Opaque | Variable | Uninterpreted string |

**Object identifier**



## SNMP PDU

| *Type* | *Tag (Binary)* | *Tag (Hex)* |
|---|---|---|
| GetRequest | **10100000** | **A0** |
| GetNextRequest | **10100001** | **A1** |
| Response | **10100010** | **A2** |
| SetRequest | **10100011** | **A3** |
| GetBulkRequest | **10100101** | **A5** |
| InformRequest | **10100110** | **A6** |
| Trap (SNMPv2) | **10100111** | **A7** |
| Report | **10101000** | **A8** |

## DNS Packet Format