Internet Networking Homework 1

Yosef Goren

May 2, 2023

Contents

1 Special-Use IPv4 Addresses ; `rfc3300`

1.1 Private Network Addresses

3 Address spaces are mentioned as ones dedicated to private networks are mentioned in the document:

1. `10.0.0.0/8`
   A private network using this adress space might have up to 2^{32-8} devices.
2. `172.16.0/12` - Max devices: 2^{32-12}.
3. `192.88.0/16` Max devices: 2^{32-16}.

1.2 Contemporary usage of private network address spaces

The private networks are now mainly used for increasing the amount of devices that could be put in a single adress space - by using the private address spaces in conjunction with NAT; This enables the use of a single IPv4 address for many devices which are all in the 'private network' of the gateway.

On the one hand - this solution makes networks that use it much more compilcated and confusing to configure.
On the other hand this likely decreases the amount of public IPv4 addresses in use by an order of magnitude, which not only helps us keep using IPv4 with more devices than there are adresses - but more importantly helps keep routing tables smaller.

2 Subnet Addressing in IPv4

2.1

a. The maximum amount of hosts is 2^{32-20}

b. In the Original Classful schemes, all addresses that start with a value between 128 and 191[1], are class B addresses - including this one.

2.2

The address `78.12.100.0/21` only has only 11 bits allocated to host suffix, so addresses inside it must match it in the first 21 bits.
In binary form:

```
_. 01001110.00001100.01100100.00000000 (78.12.100.0)
---------------------><-----------
a. 01001110.00001100.01100100.00001110 (78.12.100.14)
b. 01001110.00010101.01100100.00000001 (78.21.100.1)
c. 01001110.00001100.00000000.00000001 (78.12.0.1)
d. 01001110.00001100.01100000.00000000 (78.12.96.0)
e. 01001110.00001100.01101100.00000000 (78.12.108.0)
```

After looking carefully, we see that only a,d and e match in the first 21 bits.

[1] Corresponding to `10******`

a. `78.12.100.14` - in network.

b. `78.21.100.1` - not in network.

c. `78.12.0.1` - not in network.

d. `78.12.96.0` - in network.

e. `78.12.108.0` - in network.

2.3

a. The subnet mask: `255.255.192.0`. The network addresses:

- `130.62.0.0/18`
- `130.62.64.0/18`
- `130.62.128.0/18`
- `130.62.192.0/18`

b. The subnet mask: `255.255.252.0`. The network addresses:

- `130.62.0.0/22`
- `130.62.4.0/22`

c. The subnet mask: `255.255.252.0`. The network addresses:

- `130.62.0.0/18`
- `130.62.4.0/18`
- `130.62.8.0/18`
- ...
- `130.62.252.0/18`

3 Routing Configuration Utilities Overview

3.1 ARP cache

a. My connection interface is:

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : technion.ac.il
   IPv4 Address. . . . . . . . . . . : 192.168.59.127
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Default Gateway . . . . . . . . . : 192.168.56.1
```

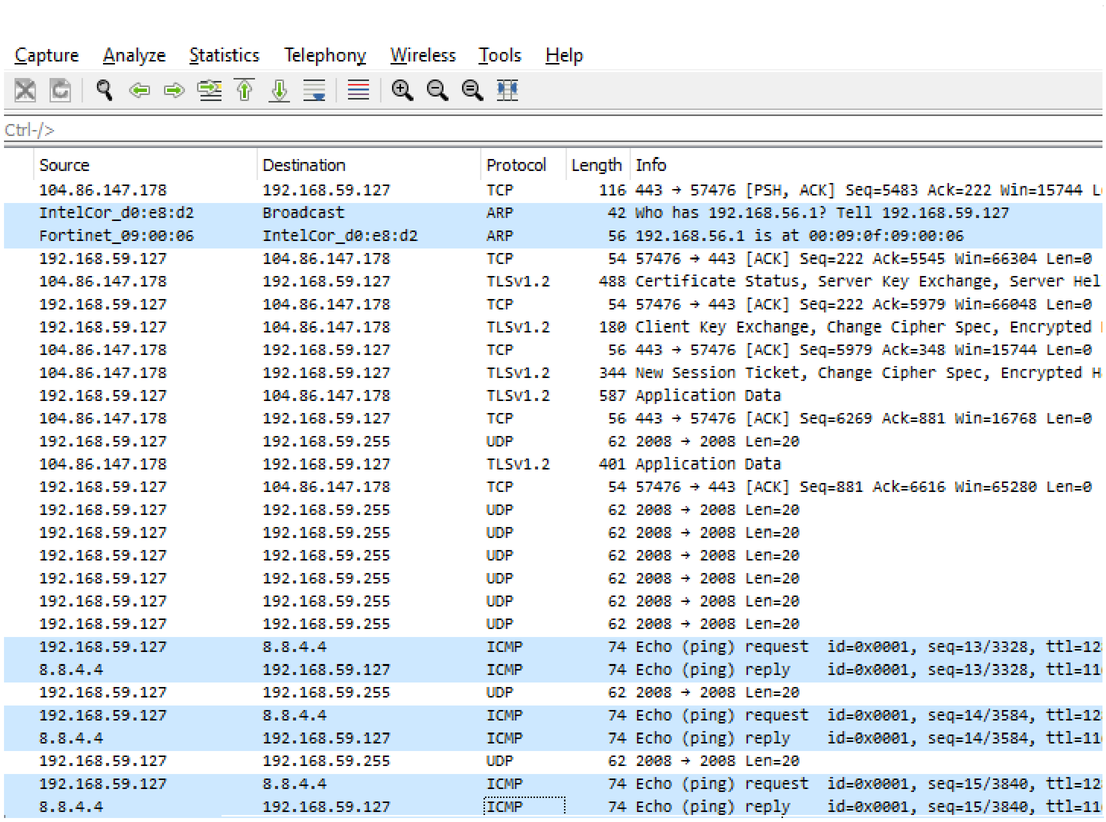
And the ARP cache contains:

```
C:\Users\pc>arp -a 192.168.56.1

Interface: 192.168.59.127 --- 0x7
Internet Address      Physical Address      Type
192.168.56.1          00-09-0f-09-00-06     dynamic
```

so `00-09-0f-09-00-06` is the MAC address of my default gateway.

b. All of the relevant packets have been highlighted.
The first two highlighted packets are from the ARP protocol.
The rest of the highlighted packets are for the ICMP protocol itself.



Source	Destination	Protocol	Length	Info
104.86.147.178	192.168.59.127	TCP	116	443 → 57476 [PSH, ACK] Seq=5483 Ack=222 Win=15744 L
IntelCor_d0:e8:d2	Broadcast	ARP	42	Who has 192.168.56.1? Tell 192.168.59.127
Fortinet_09:00:06	IntelCor_d0:e8:d2	ARP	56	192.168.56.1 is at 00:09:0f:09:00:06
192.168.59.127	104.86.147.178	TCP	54	57476 → 443 [ACK] Seq=222 Ack=5545 Win=66304 Len=0
104.86.147.178	192.168.59.127	TLSv1.2	488	Certificate Status, Server Key Exchange, Server Hel
192.168.59.127	104.86.147.178	TCP	54	57476 → 443 [ACK] Seq=222 Ack=5979 Win=66048 Len=0
192.168.59.127	104.86.147.178	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted
104.86.147.178	192.168.59.127	TCP	56	443 → 57476 [ACK] Seq=5979 Ack=348 Win=15744 Len=0
104.86.147.178	192.168.59.127	TLSv1.2	344	New Session Ticket, Change Cipher Spec, Encrypted H
192.168.59.127	104.86.147.178	TLSv1.2	587	Application Data
104.86.147.178	192.168.59.127	TCP	56	443 → 57476 [ACK] Seq=6269 Ack=881 Win=16768 Len=0
192.168.59.127	192.168.59.255	UDP	62	2008 → 2008 Len=20
104.86.147.178	192.168.59.127	TLSv1.2	401	Application Data
192.168.59.127	104.86.147.178	TCP	54	57476 → 443 [ACK] Seq=881 Ack=6616 Win=65280 Len=0
192.168.59.127	192.168.59.255	UDP	62	2008 → 2008 Len=20
192.168.59.127	192.168.59.255	UDP	62	2008 → 2008 Len=20
192.168.59.127	192.168.59.255	UDP	62	2008 → 2008 Len=20
192.168.59.127	192.168.59.255	UDP	62	2008 → 2008 Len=20
192.168.59.127	192.168.59.255	UDP	62	2008 → 2008 Len=20
192.168.59.127	192.168.59.255	UDP	62	2008 → 2008 Len=20
192.168.59.127	8.8.4.4	ICMP	74	Echo (ping) request id=0x0001, seq=13/3328, ttl=12
8.8.4.4	192.168.59.127	ICMP	74	Echo (ping) reply id=0x0001, seq=13/3328, ttl=11
192.168.59.127	192.168.59.255	UDP	62	2008 → 2008 Len=20
192.168.59.127	8.8.4.4	ICMP	74	Echo (ping) request id=0x0001, seq=14/3584, ttl=12
8.8.4.4	192.168.59.127	ICMP	74	Echo (ping) reply id=0x0001, seq=14/3584, ttl=11
192.168.59.127	192.168.59.255	UDP	62	2008 → 2008 Len=20
192.168.59.127	8.8.4.4	ICMP	74	Echo (ping) request id=0x0001, seq=15/3840, ttl=12
8.8.4.4	192.168.59.127	ICMP	74	Echo (ping) reply id=0x0001, seq=15/3840, ttl=11

- The purpose of the first packet isto find out what is the physical (MAC) address of the default gateway. This is required since at that point - the arp cache at the gateway's entry was empty since we have just deleted it.
- The second packet is a response to the first one (Still ARP protocol), and it specifies the value of the requested address.

Without this preceeding ARP interaction - the host would not know how to interact with the default gateway.

- The next 6 packets are a part of the ICMP protocol. The first one requests a response from the host at `8.8.4.4`, and the next packet is the response from `8.8.4.4` back to the requester.
  The third packet is also a request for response and the next on is the response, and so on.

c. If we were to run the `ping 8.8.4.4` command again - we would not see the same packets transfered - since there would be no need for the ARP packets this time, the IP to MAC address mapping for the default gateway is already cached and there is no need to querry for it again.

d. Sending `ping 8.8.4.4l 4000` results with:

```
IntelCor_d0:e8:d2       ARP      56 Who has 192.168.59.127? Tell 192.168.56.1
Fortinet_09:00:06       ARP      42 192.168.59.127 is at e4:02:9b:d0:e8:d2
192.168.59.255          UDP      62 2008 → 2008 Len=20
8.8.4.4                 IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=9c30) [Reassembled in #19]
8.8.4.4                 IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=9c30) [Reassembled in #19]
8.8.4.4                 ICMP   1082 Echo (ping) request  id=0x0001, seq=17/4352, ttl=128 (no response found!)
192.168.59.255          UDP      62 2008 → 2008 Len=20
192.168.59.255          UDP      62 2008 → 2008 Len=20
192.168.59.255          UDP      62 2008 → 2008 Len=20
192.168.59.255          UDP      62 2008 → 2008 Len=20
192.168.59.255          UDP      62 2008 → 2008 Len=20
8.8.4.4                 IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=9c31) [Reassembled in #27]
8.8.4.4                 IPv4   1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=9c31) [Reassembled in #27]
8.8.4.4                 ICMP   1082 Echo (ping) request  id=0x0001, seq=18/4608, ttl=128 (no response found!)
192.168.59.255          UDP      62 2008 → 2008 Len=20
```

(a) Each ICMP request was split into three fragments, This is since the MTU for the system is 1500 bytes, meaning at-least 3 fragmets are required to contain a 4000-byte message.

(b) IPv4 Header details:

Checksum	Fragment Offset	Flags	Id	Total Length	#Fragments
0x70bd	0	001	0x9c30	1500	1
0x7004	1480	001	0x9c30	1500	2
0x90fb	2960	000	0x9c30	1068	3

(c) A service that supports fragmeted ICMP exposes itself to DOS attacks.
This is since fragmeted packets require the reciver to save a state (and allocate resources) for each sender; moreover - unlike a regular http session - ICMP is not built on TCP, and so no 3 way TCP handshake was used to authenticate the sender IP address - which means IP spoofing can be used in a potential DOS attack.

(d) `C:\WINDOWS\system32>ping localhost -l 4000`

```
Pinging DESKTOP-86KTLQ1 [::1] with 4000 bytes of data:
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms

Ping statistics for ::1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

3.2 `tracert`

a. The type field value is 8, which means it's a ping request.

b. The responses have type 11, which means the request packet has exceeded it's time-to-live value.

c. `tracert` works as follows:

1. `tracert` sends a series of ICMP packets, each with an increasing `TTL` (Time To Live) value, starting from 1.
2. The first packet with `TTL=1` is sent to the target IP address. When the packet reaches the first router (the router closest to my computer), the `TTL` value is decremented by 1, and the router sends an `ICMP Time Exceeded` message back to my computer.
   Traceroute might also repeat each step multiple times to verify that the routing through a specific node is consistent; in the case of our experiment - it repeats each step 3 times.
3. The `tracert` command records the IP address of the router that sent the Time Exceeded message, along with the round-trip time it took for the packet to travel to and from that router.
4. The `tracert` command then sends another ICMP packet with a `TTL` value of 2. This packet will reach the first router and be forwarded to the next router along the path to the target IP address.
5. The process repeats, with the `TTL` value increasing by 1 each time until the target IP address is reached or a maximum number of hops is reached.

d. There are a few reasons why this method of detecting the routing path might yield false results: firstly - the routing path might change between packet to packet depending on how routers on the way might decide the optimal routing is at that moment. Moreover - some routers on the way might not respond to the timed-out ICMP packets the way we expect. For example - some might give a replay after dropping a timed-out packet.

Source	Destination	Info
192.168.70.73	18.172.153.35	Echo (ping) request id=0x0001, seq=50/12800, ttl=1 (no response found!)
192.168.68.1	192.168.70.73	Time-to-live exceeded (Time to live exceeded in transit)
192.168.70.73	18.172.153.35	Echo (ping) request id=0x0001, seq=51/13056, ttl=1 (no response found!)
192.168.68.1	192.168.70.73	Time-to-live exceeded (Time to live exceeded in transit)
192.168.70.73	18.172.153.35	Echo (ping) request id=0x0001, seq=52/13312, ttl=1 (no response found!)
192.168.68.1	192.168.70.73	Time-to-live exceeded (Time to live exceeded in transit)
192.168.70.73	18.172.153.35	Echo (ping) request id=0x0001, seq=53/13568, ttl=2 (no response found!)
132.68.231.41	192.168.70.73	Time-to-live exceeded (Time to live exceeded in transit)
192.168.70.73	18.172.153.35	Echo (ping) request id=0x0001, seq=54/13824, ttl=2 (no response found!)
132.68.231.41	192.168.70.73	Time-to-live exceeded (Time to live exceeded in transit)
192.168.70.73	18.172.153.35	Echo (ping) request id=0x0001, seq=55/14080, ttl=2 (no response found!)
132.68.231.41	192.168.70.73	Time-to-live exceeded (Time to live exceeded in transit)
192.168.70.73	18.172.153.35	Echo (ping) request id=0x0001, seq=56/14336, ttl=3 (no response found!)
132.68.231.54	192.168.70.73	Time-to-live exceeded (Time to live exceeded in transit)
192.168.70.73	18.172.153.35	Echo (ping) request id=0x0001, seq=57/14592, ttl=3 (no response found!)
132.68.231.54	192.168.70.73	Time-to-live exceeded (Time to live exceeded in transit)

Figure 1: The first 6 packets seen have identified the first router in the path to `www.walla.co.il`, each pair is a ping request, followed by a response from the router at the distance that equals to the value of the `TTL` (which is 1), and the interaction is repeated 3 times. The following 6 packets are used to identify the next router in the path - at the distance of 2, and so on.

bootp

Source	Destination	Protocol	Length	Info
10.0.0.18	10.0.0.138	DHCP	342	DHCP Release - Transaction ID 0xa1eff8a8
0.0.0.0	255.255.255.255	DHCP	346	DHCP Discover - Transaction ID 0x3b795289
10.0.0.138	10.0.0.18	DHCP	380	DHCP Offer - Transaction ID 0x3b795289
0.0.0.0	255.255.255.255	DHCP	389	DHCP Request - Transaction ID 0x3b795289
10.0.0.138	10.0.0.18	DHCP	380	DHCP ACK - Transaction ID 0x3b795289

Figure 2: The DHCP messages include the first one from previously releasing the allocated address, followed by Discover, Offer, Request and ACK messages.

4 DHCP ; `RFC 2131`

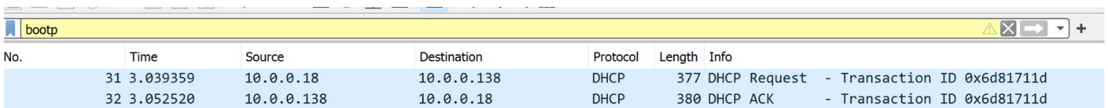
4.1 Wireshark Analysis

The DHCP server needs to know the MAC address since it's the only way it can identify the client (if it already had an IP it would not need DHCP!), moreover - just because the MAC address appears in the packet does not mean it gets to the DHCP server as layer 2 headers (including the MAC address) are not passed on to the protocols above them.

4.2

It is the same address. This means that the client and server are directly connected at layer 2, and that no DHCP relay is required for the two to communicate.

4.3



bootp

No.	Time	Source	Destination	Protocol	Length	Info
31	3.039359	10.0.0.18	10.0.0.138	DHCP	377	DHCP Request - Transaction ID 0x6d81711d
32	3.052520	10.0.0.138	10.0.0.18	DHCP	380	DHCP ACK - Transaction ID 0x6d81711d

No new DHCP messages were sent during the 10 minutes.
This means that the lease time is more than 20 minutes - since after half the lease time has expired - the client is meant to send a DHCP Request message to renew it's lease.
The lease time is dictated by the DHCP server, it's timing can be controled by it's configuration, or the lease time requested.
When deciding what the lease time should be - there is a trade-off: have a lease time which is too long might mean a-lot of unused addresses are still allocated to hosts which did not relase them properly when they where done using them, and on the other hand - having too short of a lease time will cause unnssecery traffic that might take bandwidth that could otherwise be used for something else.

4.4

a. In it's DHCP request, the server will use option number 61, type 1; and provide `192.168.32.32` as it's requested address.

b. The server will specify options 3, 6 and 1 in it's request.

c. The server should specify option 51, and provide the lease time value in seconds (so the number of seconds in a week).

5 OSPF ; `RFC 2328`

5.1

In the context of the Directed Graph data scructure saved for the OSPF protocol, each node represents a router or a whole Local-Area-Network, while an arch represents a physical-layer-connection between two nodes s.t. information will be forwarded in the direction of the arch.
For example router-router connection means that the two routers will forward information to eachother (depending on the arch direction) at layer 2, and a router to LAN connection means that the router will forward information to the LAN.

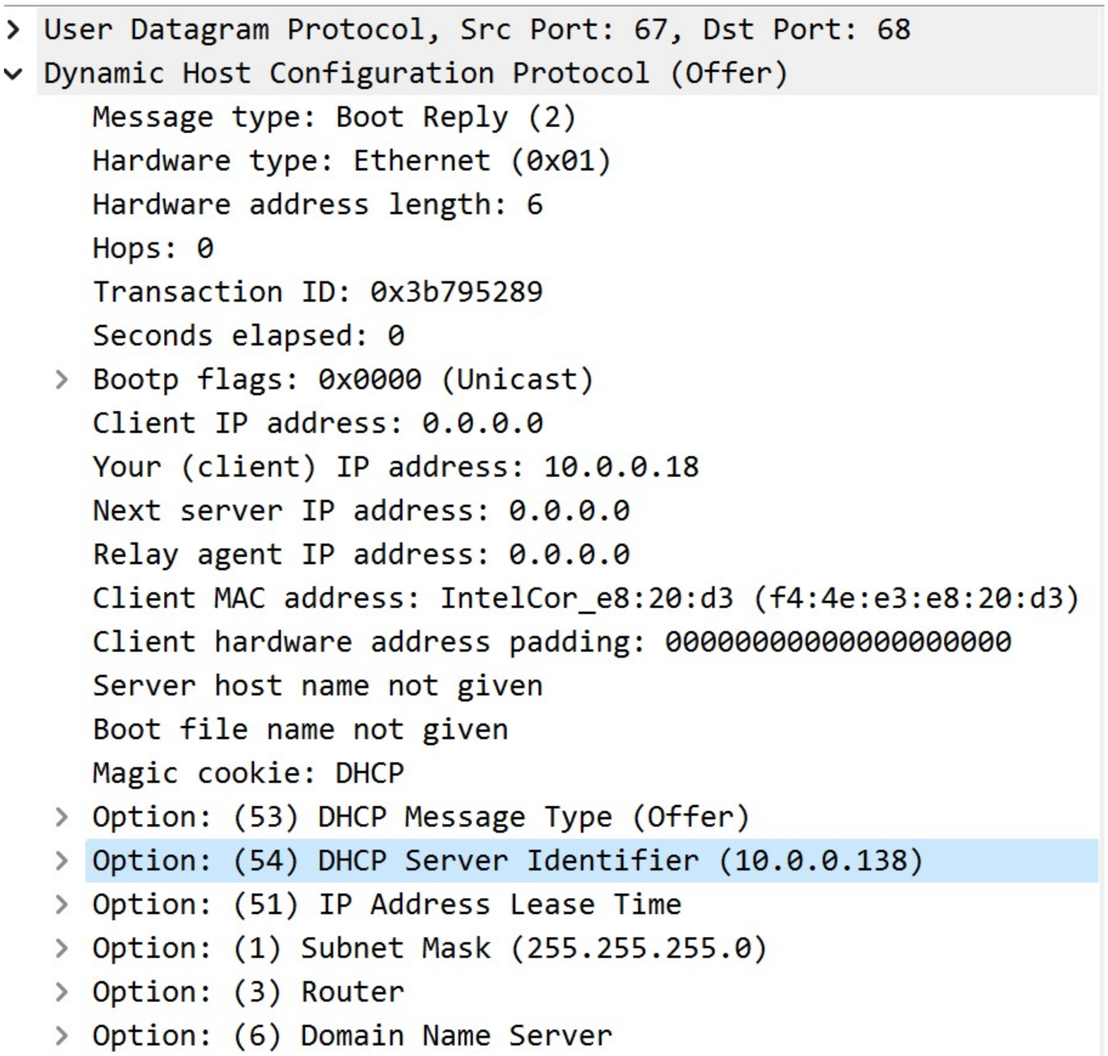


Figure 3: As can be seen, the DHCP server address is the same address that is sending the DHCP server messages back to the client.

5.2

Since a stub network is only connected to that one router - there is no possiblity that the stub network will every forward any information from a different network through itself and into it's gateway router.

5.3

The main difference between Brodcast networks and NBMA networks is that the latter do not support brodcasting messages.
This also means that if node A in the DHCP graph is from outside an NBMA network with multiple is connected to node B inside tehe NBMA network - it might still not be able to directly communicate with another router C which is also inside the same NBMA network; this means that the router B will likely be required to forward information between routers A and C - and hence the connection to it should be bi-directional.

5.4

A 'point-to-point' connection is a direct connection between two routers - by a single packet transfer on the underlying layer 2 protocol.
Such a connection will be saved as a 'type 1' (point-to-point) link descriptor, with link ID equal to the IP of the neighboring router.
Additionally - such a connection will create another 'type 3' link desciptor, which represents a stub link, similarly with the router's IP as the ID.

5.5

Large AS's are often divided into distinct regions, and the routers in each region are only required to hold the relavant data structures to the routers in their own regions.
When inter-regional routing is needed - the routing is done through what's called the 'backbone' of the OSPF network; this backbone is one of the regions that comprize the AS and it essentially acts as a hub for the AS through which the inter-regional communications is transmitted.

5.6

The five message types that exist in OSPF are:

1. Hello: These messages are essentially ping messages used to discover and maintain who the router's neighbors are.

2. Database Description: These messages summerize the content of the database saved in the sending router, which enables others to update their own database accordingly.

3. Link State Request: These messages are requests to recive other's databases.

4. Link State Update: These messages request other's to update their own databases.

5. Link State Ack: These message indicate that a required database update has been completed. The use of these acknolagments helps improve the reliability of the protocol in cases of unreliable connections or crashes.