TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI

VICTORIA

UNIVERSITY OF WELLINGTON

NWEN 302

Lab 2

Wael Aldroubi
300456658

Introduction:

Computers do communicate and exchange information as zeros and ones, but to process different kind of data they are using unique mechanism, we have seven layers to communicate each layer is responsible about specific job, the most famous two are Network layer (Routers understand that layer) and Data link layer (Switches understand that layer).

On the network when a device want to communicate with other device on the same local network will ask for the (Mac address)[1] of that device, and will get it using the local (IP address)[2].

Mac address is at the layer two which is data link layer, known and understood by the switch.

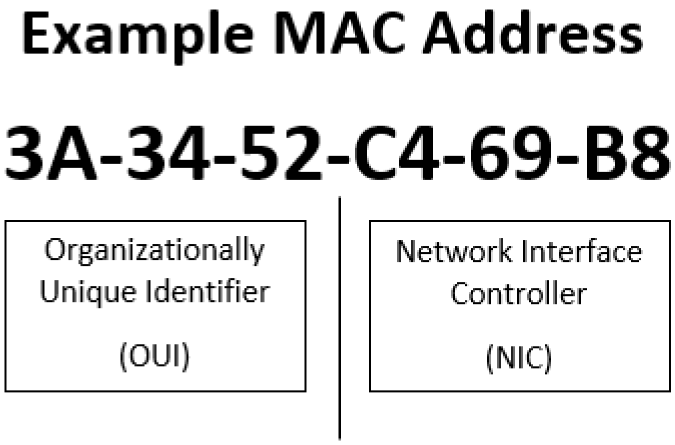
IP address is at the layer three which is Network layer, known and understood by the router.

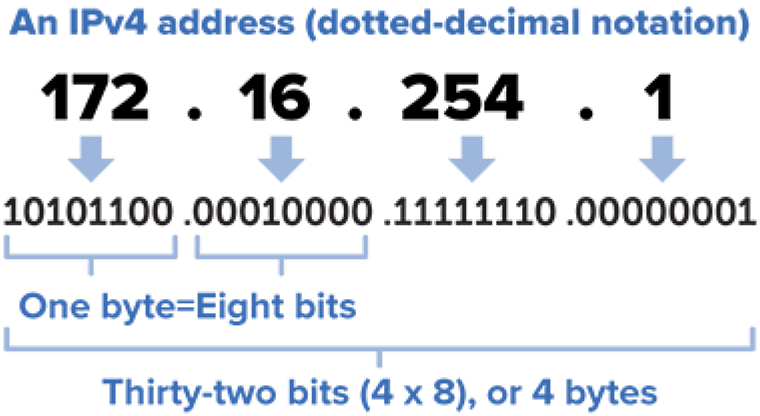
When a device wants to communicate with the other one in IPV4[3], will send ARP[4] (Address resolution protocol) to find that device's MAC address suing the IP address.

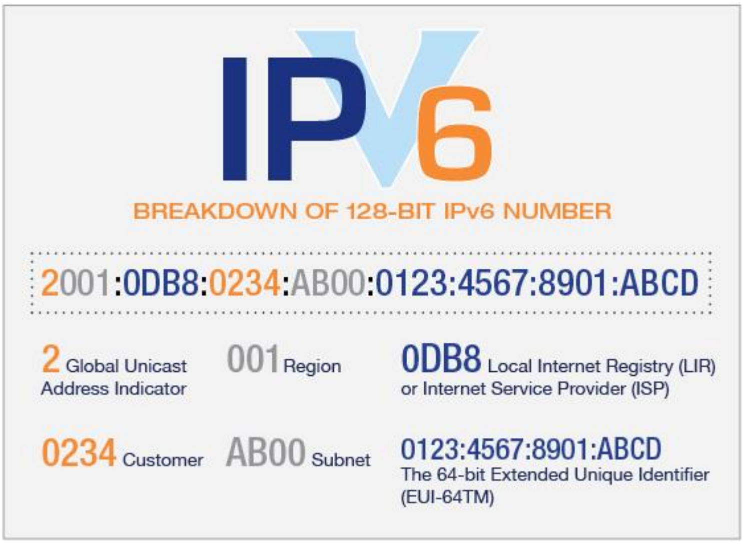
IP is an address assigned to each device on the network as an ID to find that device for communication.

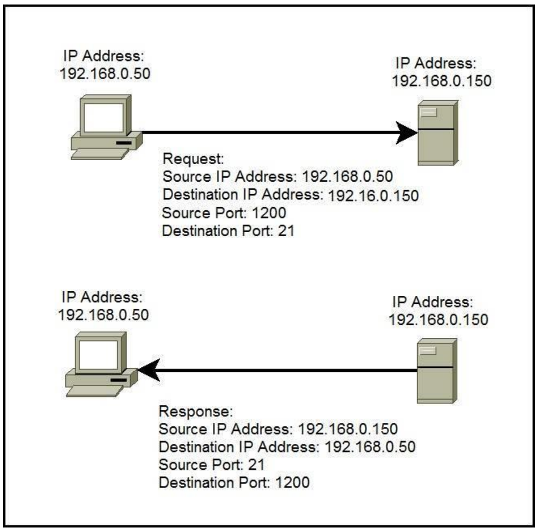
If the device has IPV6, then it will use the NDP[5] (Neighbour discovery protocol), to send something similar to ARP to discover the routers on the network then to discover neighbour devices on the same network.

In this Lab, I am modifying code represent transferring data through the seven layers schema, and implementing code to figure the type of data and wither it is IPV4 or IPV6, to implement Address resolution protocol and store it in a list for a period of time, or Neighbour discovery protocol to store routers and nearby devices on the local network.


Example MAC Address
3A-34-52-C4-69-B8
Organizationally Unique Identifier (OUI)
Network Interface Controller (NIC)


An IPv4 address (dotted-decimal notation)
172 . 16 . 254 . 1
10101100 .00010000 .11111110 .00000001
One byte=Eight bits
Thirty-two bits (4 x 8), or 4 bytes


IPV6
BREAKDOWN OF 128-BIT IPv6 NUMBER
2001:0DB8:0234:AB00:0123:4567:8901:ABCD
2 Global Unicast Address Indicator
001 Region
0DB8 Local Internet Registry (LIR) or Internet Service Provider (ISP)
0234 Customer
AB00 Subnet
0123:4567:8901:ABCD The 64-bit Extended Unique Identifier (EUI-64TM)

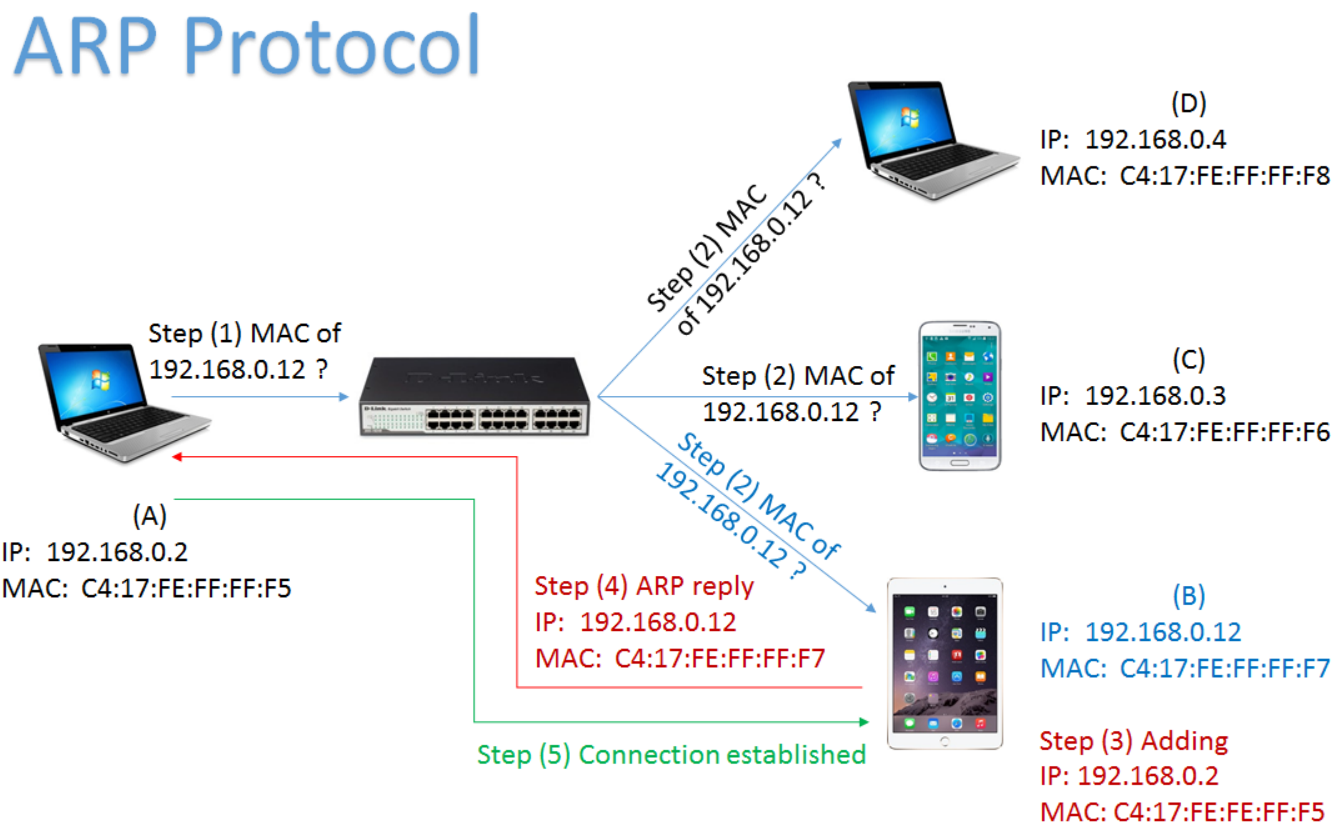

IP Address: 192.168.0.50
IP Address: 192.168.0.150
Request:
Source IP Address: 192.168.0.50
Destination IP Address: 192.16.0.150
Source Port: 1200
Destination Port: 21
IP Address: 192.168.0.50
IP Address: 192.168.0.150
Response:
Source IP Address: 192.168.0.150
Destination IP Address: 192.168.0.50
Source Port: 21
Destination Port: 1200

Examples of:
MAC address, IPV4 address, IPV6 address and how IP address transferred through the network.

ARP (Address resolution protocol) explained:

Address resolution protocol is used in network layer protocol to convert local IP address to MAC address, such as Ethernet address, to make communication between devices.

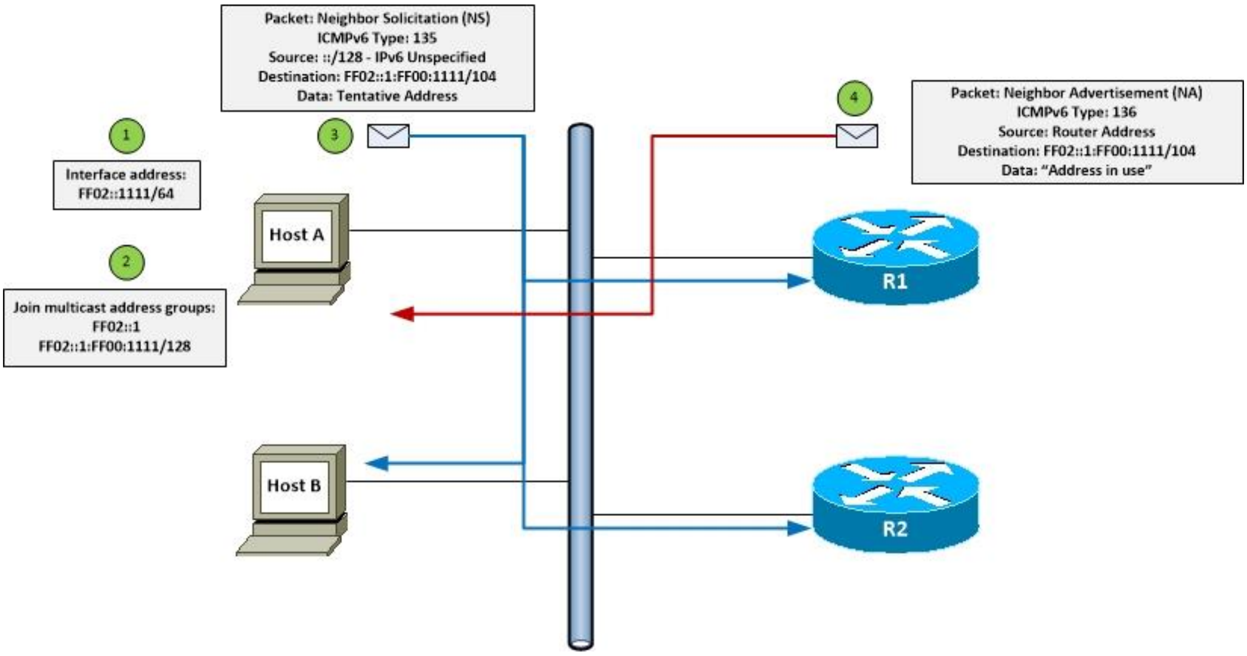
The source device that wants to communicate with destination device will send a broadcast message called ARP, asking for MAC address of a specific IP address of the destination device, then all devices will receive it, and the target device will replay with his MAC address to start data transaction process.



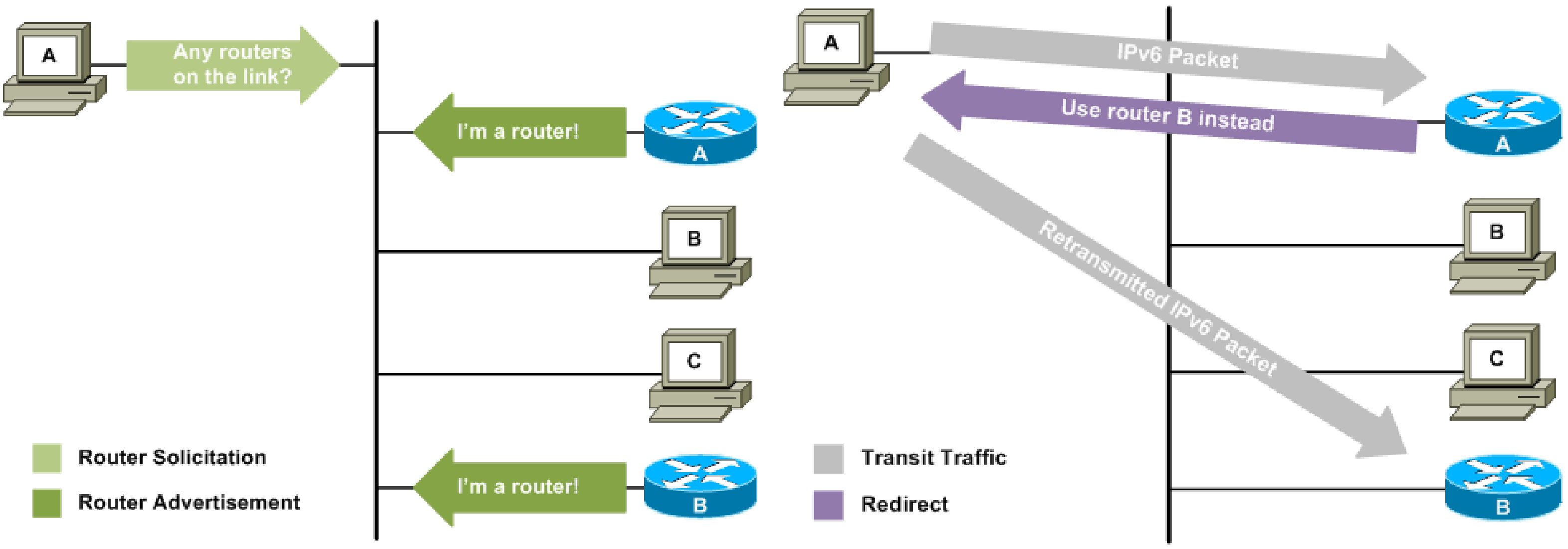
Example of ARP protocol and how it works.

NDP (Neighbour discovery protocol) explained:

1. Router solicitation:
   one device send router solicitation request to find all available routers on the network for later stage to find the nearest router to the target device.
2. Router advertisement:
   the router send a reply to the router solicitation request to let the device know about its existence and to add it to his list.
3. Neighbour solicitation:
   the device will send a request to the neighbour device asking for IP address to add it to its list as well, similar to router solicitation.
4. Neighbour advertisement:
   the neighbour of that device that sent the neighbour solicitation request will answer with a replay allowing that device to add it to its list.
5. Redirect message:
   is when a device communicate with a router asking for a specific device, the router will advise the device to communicate with another router near that device, for faster handling of data communication.



Packet: Neighbor Solicitation (NS)
ICMPv6 Type: 135
Source: ::/128 - IPv6 Unspecified
Destination: FF02::1:FF00:1111/104
Data: Tentative Address
1
Interface address:
FF02::1111/64
2
Join multicast address groups:
FF02::1
FF02::1:FF00:1111/128
3
4
Packet: Neighbor Advertisement (NA)
ICMPv6 Type: 136
Source: Router Address
Destination: FF02::1:FF00:1111/104
Data: "Address in use"
Host A
Host B
R1
R2



The code:

For Part A:

The provided code is a representation of some of the OSI layers model, and I have to make a huge modification to Network layer and link layer, to allow devices instead of hard code of the table, to store them dynamically.

No changes to data application which represent the application layer, to process the data and decide what kind of IP it does require.

Link layer, changes will applied to accept packet and send packet functions, responsible of either accept sent data if it's meant to be sent to our device, or process to send data to target device after deciding what type it is.

No changes made to the main application as it is its job to process traffic transaction and to record the list of neighbours.

Network layer, is where the most of modification was implemented, where to handle

1. Conversion from IPV4, IPV6 addresses to Mac address.
2. To accept packet if it meant to be sent to me, or drop it if it is not for me in case of IPV4.
3. Will process neighbour solicitation and neighbour advertisement in case of IPV6.
4. To handle sending data request, to decide if it is ARP, IPV4 or if it is router or neighbour solicitation in case of IPV6.
5. Creates ARP request and ARP reply.
6. Control ICMP (The internet Message protocol) messages services.

Preferences, just changed the number of hosts to 4 hosts.

For transport layer, nothing changed, this layer is responsible of managing and organizing the transferred data between source and destination layers, like handle data send acknowledgement and receive it, source and destination addresses.

For Part B:

In every switch there is mechanism to take coming traffic and sending it to target device, the switch will have a table storing all devices information like the IP addresses, and when receive a new request and the list of requests is full, it will traverse the list and choose the one with the shortest time to live and replace it.

This function will check if the MAC address of the requested device is on the list, if it is there will process it and complete the request, otherwise it will send a request asking for it depending on the previously explained mechanisms.

References

github. (n.d.). *github*. Retrieved from github: https://github.com/mininet/mininet/wiki/Introduction-to-Mininet

github. (n.d.). *github*. Retrieved from github: https://github.com/shen79/ipfu

IP. (n.d.). *youtube*. Retrieved from youtube: https://www.youtube.com/watch?v=a1AQfjWwPaE

IP2. (n.d.). *youtube*. Retrieved from youtube: https://www.youtube.com/watch?v=Y6PqkMBenQU

IPV6. (n.d.). *cisco*. Retrieved from cisco: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6/configuration/15-2mt/ip6-15-2mt-book/ip6-neighb-disc.html

python. (n.d.). *python*. Retrieved from python: https://python-arptable.readthedocs.io/en/latest/readme.html

python. (n.d.). *stackoverflow*. Retrieved from stackoverflow: https://stackoverflow.com/questions/22687940/python-ryu-handling-packets-using-a-switch-after-flow-was-added-to-switch

Scapy. (n.d.). *null-byte.wonderhowto*. Retrieved from null-byte.wonderhowto: https://null-byte.wonderhowto.com/how-to/build-arp-scanner-using-scapy-and-python-0162731/

Spoofer. (n.d.). *github*. Retrieved from github: https://github.com/AnisJokerDz/ARP-Spoofer

switch, F. (n.d.). *github*. Retrieved from github: https://github.com/internap/fake-switches

trick, a. s. (n.d.). *yamakira*. Retrieved from yamakira: https://yamakira.github.io/art-of-packet-crafting-with-scapy/network_attacks/arp_spoofing/index.html

youtube. (n.d.). *youtube*. Retrieved from youtube: https://www.youtube.com/watch?v=A3LFt7CHpgs