NETSEC VT2022

Practical Laboratory

Lab #1

10 February 2022

Group 40:

Jawdat Kour (jako0237)

Anas Kwefati (ankw6718)

Munish Sharma (mush5681)

Department of Computer and Systems Sciences

Spring 2022

Stockholm University

Ethernet frame forwarding

Question 1

On your computer, list the ethernet (or wireless) adapters. How many are there? What are their MAC addresses? Based on the MAC addresses can you identify the vendor of the network cards?

Running the command `ipconfig /all` on windows command prompt shows only one NIC/ Ethernet adapter on the computer since it's a desktop connected directly to the LAN network and has no wireless interface card.

The ethernet adapter has a physical address of 48 bit: `F8-BC-12-AA-1F-F4`.

Using an online tool such as https://www.ipchecktool.com/ which checks the first 24 bits (6 characters) of the MAC address, we could identify the manufacturer of the network card which is `Dell inc`. The first three octets of the Mac address are called Organization Unique Identifier (OUI) assigned to Dell by the Institute of Electrical and Electronics Engineers IEEE [1].

Question 2

In which layer of the OSI model does Ethernet switches typically operate?

The Ethernet switches operate at the Data link layer (Layer 2) of the OSI model. By maintaining a CAM table that maps the physical ports to the Mac address to the connected devices, the switch creates separate collision domains for each port. Since switches rely on MAC addresses to forward frames, there is no need to work on the Network layer (Layer 3) which deals with IP addresses and is the layer that the routers work at [2].

Question 3

Bring up your interface statistics, how many frames have been sent?

The activity area of the Ethernet interface status shows that 55 794 343 bytes of data have been sent.

Question 4

Can you find out which other MAC addresses your computer has communicated with in the recent past? (Hint: ARP-cache)

The Address resolution protocol (ARP) is used to discover the MAC address associated with the IP address. Operating systems such as windows cache the findings in an ARP table.

The command `arp -a` on windows command prompt displays all the cached ARP entries in a table that contains: IP address | Physical address | Type of entry (static / dynamic).

Question 5

View the ethernet frame part of the sent/received packets. How many frames are listed? Can you correlate them with the sent ping ip packets? What is the receiving MAC address? Is that the MAC address of the www.su.se server? (If not, what kind of device is it, and who manufactured it?)

Verify that the frames are sent with the MAC address of your computer's active interface (i.e. that the sending MAC address is the same as the interface you think you're using).

Wireshark shows that four ICMPv4 request packets have been sent and four ICMPv4 reply packets have been received when pinging from windows command prompt to the www.su.se server that has the IP address (`193.11.30.171`). In total 8 ICMP frames are listed.

By looking at the Ethernet part of Packet details in wireshark, we recognize two MAC addresses. The first MAC is the NIC address of the active computer that sends the ICMP requests (our computer in this case). The second MAC address is (`e0:b9:e5:d5:ba:ba`), which is of the source that send the ICMP reply packets, this MAC came from 'Technicolor Delivery Technologies Belgium NV', which seems kind of gateway/firewall that stands in front of the web server.

Network Layer packet forwarding

Question 6

What does the above (sudo ping -l 51020 -f -s 51020 -a 127.0.0.1) ping command do?

Which network protocol does the ping utility use to craft the ping request and response packets? On which layer of the TCP/IP stack does this network protocol belong?

The previous ping command allows to execute a ping command as a super user and send ICMP ECHO_REQUEST packets to the local host with the following parameters :

`-l 51020` sends `51020` packets as fast as possible before falling into its normal mode of behaviour.

`-f` Outputs packets as fast as they come back. print "." For every ECHO_REQUEST sent, and print "backspace" while for every ECHO_REPLY received. This provides a rapid display of how many packets are being dropped.

`-s 51020` the packet size of data to be sent in each ICMP request (8 bytes of ICMP header is added)

`-a` Include a bell when a packet is received (Audible). Will be ignored since other parameters are present.

The **Internet Control Message Protocol** (**ICMP**) is used by ping. Two versions of the protocol are available to use with the ping utility which are ICMPv4 and ICMPv6 to check the connectivity with IPv4 address and IPv6 respectively. This Protocol works at the **network layer (layer 3)** of the OSI model.

Question 7

”What options did you use? How many bytes of data did each request carry? What was the TTL value used, and what does the TTL abbreviation stand for? What was the packet loss rate? What was the average round trip time (RTT)?”

Command:

```
Ping www.su.se
```

Output:

```
Reply from 193.11.30.171: bytes=32 time=12ms TTL=243

Reply from 193.11.30.171: bytes=32 time=14ms TTL=243

Reply from 193.11.30.171: bytes=32 time=12ms TTL=243

Reply from 193.11.30.171: bytes=32 time=12ms TTL=243


Ping statistics for 193.11.30.171:

    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

    Minimum = 12ms, Maximum = 14ms, Average = 12ms
```

Each ICMP request carries 32 bytes of data. Time To Live (TTL) is 243 hops which represents the max number of hops / IP routers that the ICMP packet can go through before being discarded.

The loss rate of packets is 0% , meaning that an ICMP reply packet has been received for each ICMP request packet sent.

The average round trip time is 12 millisecond, representing the average time needed for each ICMP packet to reach the destination plus the time of the ICMP reply packet to be received back at the source.

Question 8

”Which protocols are employed for sending a ping request from one system to another? Can you describe the order of protocol encapsulation applied according to the TCP/IP or OSI model? What are the values for the type and code fields of the ping request? What is the ID value of the IP packet? What is the type of the Ethernet frame?”

”Highlight a ping reply packet below the previously selected frame. Examine the values of all protocol fields and compare them with those of a ping request packet. Which fields have the same value between them?”

```
ping www.chalmers.se
```

Internet protocol (IPv4) is used to send the ICMP packets between the systems.

The ICMP packet is encapsulated inside the IPv4 packet at the network layer of the OSI model, which in turn is encapsulated inside the Ethernet frame at the data link layer of OSI model. The Ethernet frame is converted from digital bits to appropriate form for transmission.

Ping Request uses :

- ICMP type: `8 = Echo`
- ICMP code: `0`

IPv4 Identification field value : `0x8dfe` which is used for uniquely identifying the group of fragments of a single IP datagram

Ethernet Frame Type: `IPv4 (0x0800)`

Comparing the ICMP request packet with the ICMP reply packet, the ICMP code = 0 is still the same. However, the ICMP type is changed for the ICMP reply and it is 0 now.
These fields are also still have the same value for both ICMP request and reply packets:

- Identifier (BE): `1 (0x0001)`
- Identifier (LE): `256 (0x0100)`
- Sequence Number (BE): `9 (0x0009)`
- Sequence Number (LE): `2304 (0x0900)`

Question 9

What does ARP stand for? In which layer of the TCP/IP stack does it belong? Briefly describe its main function. Briefly describe and explain why this protocol is important in a switched network.

Examine the ARP request packet. What is the destination MAC address? Briefly explain this.

Examine the ARP reply packet. What is the source MAC address of the frame and to which network host does it belong? Does the ARP reply contain valid data?

ARP stands for Address Resolution Protocol, and is used to discover the MAC address associated with the IP address. Operating systems such as windows cache the findings in an ARP table. ARP operates on the link layer of the TCP/IP stack and it's messages are encapsulated in this layer.

When computers on the same IP subnet need to communicate, the arp request broadcast is sent within the boundary of the subnet to all other computers asking for the MAC address of a particular IP address, the computer who has the IP address will receive the request and response with ARP request telling the sender it's MAC address. The sender and the receiver will cash the ARP result for each other in their ARP tables for a while. Then the two computers can exchange the messages. Therefore This mapping between the IPs and MACs is a critical function in switching networks [3].

We couldn't find any ARP frames regarding the server. However, it was possible to capture ARP request / reply between the local router and our client machine on the same network as it is display below:

```
Technico_d5:ba:ba      Broadcast      ARP     60      Who has 192.168.1.255? Tell 192.168.1.1

Dell_aa:1f:f4  Technico_d5:ba:ba      ARP     42      192.168.1.225 is at f8:bc:12:aa:1f:f4
```

The Destination MAC address of the ARP request is : Broadcast `(ff:ff:ff:ff:ff:ff)` which is used to send the request to all hosts on the same subnet. The source MAC address of the ARP reply is the MAC of our client machine (`f8:bc:12:aa:1f:f4` in this case). The ARP reply packet doesn't contain data, just in the header fields tell the requester about the MAC address of the machine that has the particular IP address.

Transport layer

Question 10

Have the commands completed successfully? Why or why not? If a command has failed, how can the command be made to work? What does the 'l' parameter mean? (Note that the success/failure here may depend on the operating system you're using.)

Open two command shell windows and type 'nc -l 30000' in one of them and 'nc 127.0.0.1 30000' in the other. (You may have to use 'nc -l 127.0.0.1 30000' to start the server side connection.)

Also start a Wireshark session, but make sure to capture from the "loopback" interface, not your usual network interface.

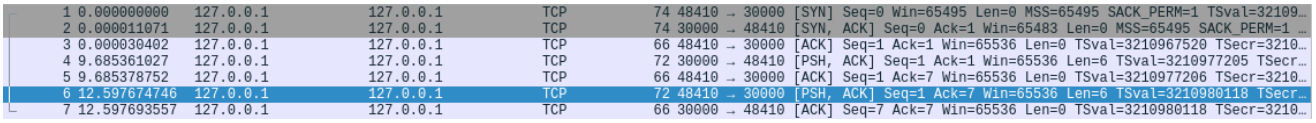
Type in one window and confirm that the text shows up in the other window. (Do this for both windows).

I have written the commands on the terminal, and it has been working successfully. The given commands allow listening on ports 21 (used by FTP) and 30000 of the computer. The -l parameter means that we are telling our system that we want to listen on the specific port for any TCP connection, or UDP activity [5]. To have a full view on how Netcat works, we need to have one that listens on port 30000, and another one that communicates and establishes a connection to this port (client: 127.0.0.1:3000). We do this, using the given commands on the assignment. I have also opened wireshark, in order to capture any packets on the network. Then, when typing text, we can detect that wireshark has captured communication packets between server/client, and the text shows in both windows. This means that we have successfully created a connection and we can communicate between the client and server.

Question 11

View the traffic in Wireshark and answer the questions: "Which transport layer protocol has been used for the communication? Provide a screenshot showing that the communication has been properly established. What are the client and server ports? How many packets flow from client to server and how many vice-versa? How many bytes are sent in each direction and in total?" Can you recover the text you sent using Wireshark? What was it?

According to Wireshark, and as can be seen in the following Figure.



The transport layer protocol that has been used is TCP. The communication has been correctly established using a 3 way handshake. The client port is 48410, whereas the server one is 30000. When communication, only 1 packet flows from client to server, and vice-versa. However, when establishing connections, we have 2 packets that went from the client, whereas only 1 packet from the server to the client. First, the client sent SYN, the server replied by SYN ACK, and then the client replied by ACK. This 3 way handshake allows to establish the connection. When communicating, (after sending the text) 72 bytes and 66 bytes have been sent in each direction, which gives us a total of 138 bytes (this contains TCP header, payload etc.). Yes, we can recover the text that has been sent using Wireshark, as we are not using any encryption. It is « Hello ».

Question 12

What were the server and client ports that were used in this communication session?

Client -> Source Port: 48410 - Destination Port: 30000
Server -> Source Port : 30000 - Destination Port : 48410

Question 13

Also, what does 127.0.0.1 mean in this context? What is special with this ip address?

127.0.0.1 is a standard local loopback IPVv4 address. It is part of the IP Network address: 127.0.0.0/8, where each address can also be used for loopback purposes (The device sends messages to itself.). This allows to test locally a network service without the need of having any physical network interface, or having to make the service available on the Internet. [6]

Question 14

Which fields and values of the captured packets give you an idea of which is the client and which is the server in each individual session?

We know which is the client, and which is the server, from the beginning when the connection is established through the 3-way handshake. It ensures that the TCP connection will not open or close, until both ends have agreed. TCP is a connection-oriented protocol, which means clients have to create a connection before any communication happens, and must end this connection, when communication has been finished. Usually communication is started by the client that wants to reach the server. Therefore, in our case, we know that the client will firstly establish the connection by sending a SYN (Synchronise), and then it will wait until the server replies with a SYN and ACK (Acknowledgement). Then, the client will reply back with an ACK, and from there connection is established, and both devices can communicate, until it is ended and agreed on by both parties. Through this we know which is

the client and which is the server, as they are using the same IP, we can use the port number to help us differentiate them. [7]

Question 15

Now we'll try to add the -u parameter to nc. Open two windows. Type 'nc -l -u 127.0.0.1 30000' in one and 'nc -u 127.0.0.1 30000' in the other. Try to communicate between the two instances as you're watching the traffic in Wireshark.

What are the differences from the TCP experiment you did above?

We are using a different Transport Layer protocol. Indeed, we are using UDP, instead of TCP. This can firstly be confirmed by the use of « -u », which is used to set the UDP mode on Netcat. Furthermore, unlike TCP, UDP is a connectionless protocol, which means that clients can send data any time with hope that it gets successfully received on the server. Indeed, UDP messages can be lost, or corrupted, as there is nothing to keep track of the data sent. During our test, and while watching traffic on Wireshark, we have not seen any pre-established connections, when sending data, it has sent it automatically. Also, UDP uses datagram packets to send and receive messages, whereas TCP uses streams to read and write packets. [7]

Application Layer

Question 16

Which application layer protocol is used? Can you identify a packet that carries the initial web request? Write down which metadata fields appear in the request and what their value means?

Can you identify the packet that carries the response? Can you identify any interesting metadata fields? Can you extract the transmitted HTML code of the visited web page? If not, why not? (Note that there is no encryption: *neverssl* is called that for a reason).

It is using HTTP protocol for the application layer. Yes, we have a packet that goes from our IP to the website's IP, where it is doing an HTTP Request named GET. This HTTP request contains the REQUEST method (in our case: GET), then the requested URI (in our case: /online), and finally the HTTP version used (here, HTTP/1.1). These values are necessary to receive the correct web page. This is called the Request Line section. Then we have the Request Headers, which contains the host (here: glowinginnercoolmelody.neverssl.com), but also the User-Agent, the Accepted Language, Encoding, and finally the Content-length.

After receiving the request, the server will reply by sending a status information with a copy of what was asked (here a webpage), to display on the browser.
The packet that carries the response comes just after, by having in the Response message header the status line, which reflects the result of the request from the server side. Here, we have a 200 OK, which means that the request has been fulfilled, and we obtain our webpage.

Yes, we get some metadata, such as the date it has been accomplished, the type of the content(text/html), the server name (here it's AmazonS3), last modified(Thursday 4 November 2021 16:34:59 GMT) but also the content length (2238 byes). Yes, we can extract the HTML code of the visited web page. As we are not using any encryption it is clearly visible in the Response body, just after the Response Header. [7, 8]

IP Addressing and subnets

In order to answer Question 17-24, the latest stable version of Ubuntu is used. A terminal emulator program Terminator and Wireshark is used to solve some problems.

Question 17

What is the lowest and the highest IP address that belong to the IP address range of the adress of your interface? What is the broadcast address of this IP address range? How many hosts can your network support?

This task is performed on a machine connected to a home network, where an IP address is assigned by an access point which is further connected to the ISP. The IP address assigned to this machine belongs to class C, which consists of 32 bits in total, where the first 24 bits are the prefix(Network ID), and the last 8 bits define the suffix (Host ID).

The machine is assigned with IP address **192.168.10.218/24,** which implies the subnet mask for this address is **255.255.255.0.** This address belongs to network **192.168.10.0/24** (254 hosts) with broadcast address **192.168.10.255.** The highest IP is **192.168.10.254/24** and the lowest IP is **192.168.10.1/24. 254 hosts in total.**

Question 18

Write down the IP address that you have been assigned. Identify the network ID of this IP address as well as the host ID of it. A technique called subnetting allows breaking a given IP address range into smaller, better manageable blocks. A subnet mask groups the network prefix along with some high-order bits from the host part into forming an 'extended' network prefix. The subnet mask is used to determine the Network ID of an IP address via the logical AND operation in the process known as "Binary AND–ing".

To solve this problem, a repetition of Question 17 is performed. To get the network ID of the IP address **192.168.10.218/24,** first we defined the prefix and suffix which are first 24 bits and last 8 bits respectively. All the prefix values were converted to the bits representation and a logical AND is performed with the machine's IP address. This process resulted in network ID **192.168.10.0/24.**

Question 19

Splitting the above IP address range in half, gives you two smaller subnets, each with a subnet mask of /25 (i.e., 255.255.255.128). What is the IP address range and the broadcast address of each subnet? How many hosts can they support each? Pick one IP

adress belonging to each subnet and write them down along with their network ID and host ID.

The IP address **192.168.10.0/24** was split into two smaller subnets with network addresses **192.168.10.0/25 & 192.168.10.128/25.** These network addresses come with 126 hosts each. In this network range, if one picks an address 192.168.10.130/25 that complies, the first 25 bits are set for the network and the rest 7 bits are rotated around to give different IP addresses for hosts.
As asked in Question, the routing table is examined on the Ubuntu machine on the terminal window using command “netstat -rn”. This command resulted in 6 columned output, starting with Destination, Gateway, Genmask, Flags, MSS Window, Irtt & Iface.
In this case the destination of the network was 0.0.0.0/192.168.10.0, gateway of network 192.168.10.1, Genmask was 255.255.255.0 (/24). Moreover, in the Flags column, two flags were set UG which indicates that a specific gateway (192.168.10.1) is needed to route traffic and another flag U which simply implies that the network is up. Initial round trip for this home network is zero, and the last column was presented with information about the Wi-Fi interface of the used machine.

Question 20

What is the IP address range of systems with which your system can communicate? Is it a private or a public IP-address? What’s the difference between these two types of addresses and what are the advantages and disadvantages of private IP-addresses?

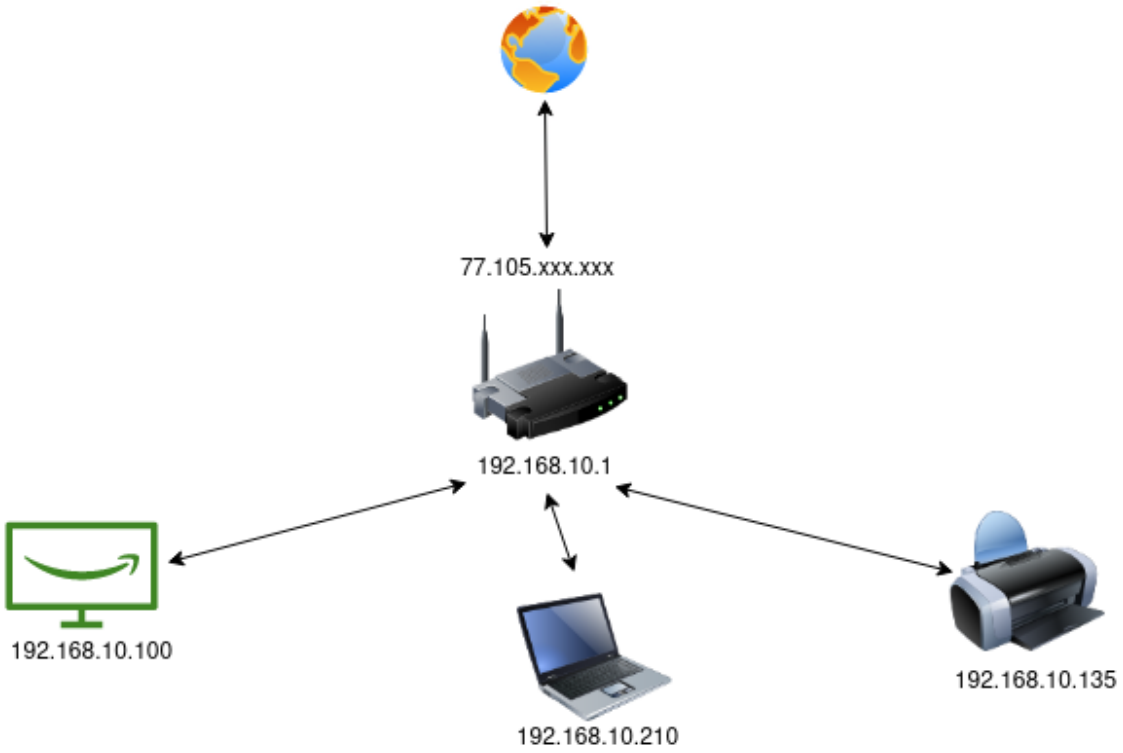
This question is partially answered in the above solutions. The IP address assigned to the current machine is a Class C (ranging between 192.168.0.0 to 192.168.255.255) of the original classful IPv4 scheme. This IP address is a private IP address which implies that it can not be routed, instead they are used to communicating within the network. Such networks need a public IP address to reach the internet. In case of IPv4, private IP addresses helped a lot, as there might not be an enough number of addresses if every single device had its own public IP address.

Basic Internetworking

Question 21

Instead list the topology of your network, identifying your host and the (typically only one) router that it is attached to.

As there was no simulator used, a topology is created using an open source diagram generator and presented below. In topology, the wireless router is connected to the internet with a public IP, and the router is acting as a DHCP server for home network providing IP addresses to the computer, printer, smart TV etc.



Question 22

What is the IP-address of the router interface you communicate with? Start Wireshark and send a ping packet to a host on the internet. Do it for long enough to invalidate the ARP-cache of your host, and capture the ARP request for your router interface.

The IP address of the router interface is 77.105.XXX.XXX, so in general when any device wants to communicate with another entity on the internet, this is the IP address used to tell the next hop, how and where to send the response back.

In order to solve the second part of the question, the host machine's ARP cache was deleted, since there are many devices connected to the home router, there was no possible way to disturb the whole network by manipulating the router's cache. So in general, when ARP cache is deleted, the machine has to ask all the devices to find a way to go out to the internet, where it gets first the gateway MAC address to communicate with. Then all the packets (ping 8.8.8.8) are sent to the gateway where the router maps the requested IP to the ARP table entry and sends the packet to the next hop.

Question 23

Study the capture. Note the recipient MAC-address of the ARP packet you are sending to the network. What address is that? What's special about it? Why does it work this way? (i.e. how does your computer find the address of a host that can answer ARP requests? It can't of course use an IP-address, as it doesn't know that yet...) You may have to try a few times to get a MAC-address that is not specific to a computer...

After deleting the ARP-cache on the host machine, it sends an ARP request to all devices connected on the network with MAC address to all the bits set (FF:FF:FF:FF:FF:FF). This sort of address is used so that all machines accept the request and then control the IP address coming along the request, if matches then an ARP response is sent otherwise the frame is rejected.

Question 24

Describe briefly what the difference is between a host that receives an IP-packet with a destination address that is not its own, and what a router does when it receives an IP-packet on an interface where that address does not belong to the router. (As is the case here). Hint: look up the term "forwarding".

The difference between a host and router when it comes to ARP broadcasting is, a computer accepts the packet and looks for the destined IP address, if it is not its own address the frame is rejected and no more action is performed. On the other hand, when a router receives a broadcast frame, it looks at the destination IP address, if not its own then it looks in the cached ARP table, if there is an entry for that IP address, the router forwards the packet to the mapped MAC address otherwise it broadcast it further and on success update the ARP table.

Application Layer Protocols (DHCP and DNS)

DNS

Question 25

Find out which DNS server you're currently configured to use. Where is that server (i.e. who runs that server). Do you have one or two (or several) DNS-servers configured? Why may there be more than one?

The DNS server that we are currently configured to use is OWNIT Katarinav agen 15. This server is located in Stockholm by the company OWNIT, which is an ISP provider. I have only one DNS-server configured. There may be more than one DNS server like root, authoritative and name server to keep the redundancy intact. So in case the nearest DNS server fails to respond to the request, it can forward the request to the closest known server and the request gets resolved.

Question 26

Check the captured DNS-traffic: "What port does the DNS server use for the communication? Which transport layer protocol is used? Which flag(s) did the DNS query header have enabled and what is the Transaction ID? Of which type is the query?"

The DNS Server uses port 53 to communicate and UDP on the transport layer. It had enabled the flag 0x0100 Standard Query for the request, with a transaction ID of 0x91ee, and 0x8180 for the response, with a transaction ID of 0x91ee. The transaction ID is a 16-bits random value that had been chosen by the client [9]. The type of the query is recursive.

DHCP

Question 27

Do that and see if you can capture the DHCP request/response (if not just say that it wasn't possible). Study the DHCP request; how does the computer know which host can give it an IP-address before it can communicate on the IP-network (as it doesn't yet have an address)?

10558	128.706628	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover
10574	131.948931	192.168.1.1	192.168.1.150	DHCP	342	DHCP Offer
10576	131.951464	0.0.0.0	255.255.255.255	DHCP	362	DHCP Request
10577	131.953303	192.168.1.1	192.168.1.150	DHCP	359	DHCP ACK

Since the DHCP client doesn't know who the DHCP server is, the client broadcasts **DHCPDISCOVER** message to all hosts on the subnet by using the subnet broadcast address (`255.255.255.255`) as a destination. The DHCP client doesn't have an IP address when sending the discover message and uses 0.0.0.0 as an source IP, and its MAC address to communicate.The DHCP server receives the IP lease request and makes a lease offer by sending a **DHCPOFFER** message to the client containing MAC, IP and some other info. The client responds to the offer and broadcasts **DHCPREQUEST** message to the server, then the DHCP server sends **DHCPACK** in the final stage (Acknowledgement) to the client containing lease duration and other parameters which is used to configure the client [10].

References

[1] "MAC address - Wikipedia", *En.wikipedia.org*, 2022. [Online]. Available: https://en.wikipedia.org/wiki/MAC_address. [Accessed: 19- Feb- 2022].

[2] "Network switch - Wikipedia", *En.wikipedia.org*, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Network_switch. [Accessed: 10 Feb- 2022].

[3] "Address Resolution Protocol - Wikipedia", *En.wikipedia.org*, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Address_Resolution_Protocol. [Accessed: 10- Feb- 2022].

[5] "How to Use Netcat Commands: Examples and Cheat Sheets", *Varonis.com*, 2022. [Online]. Available: https://www.varonis.com/blog/netcat-commands. [Accessed: 19- Feb- 2022].

[6] "localhost - Wikipedia", *En.wikipedia.org*, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Localhost. [Accessed: 19- Feb- 2022].

[7] D. Comer, *Computer Networks and Internets*, 6th ed. Boston [etc.]: Pearson, 2015.

[8] "In Introduction to HTTP Basics", *Www3.ntu.edu.sg*, 2022. [Online]. Available: https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html. [Accessed: 19- Feb- 2022].

[9] "The Domain Name System — Computer Networking : Principles, Protocols and Practice", *Beta.computer-networking.info*, 2022. [Online]. Available: https://beta.computer-networking.info/syllabus/default/protocols/dns.html. [Accessed: 20- Feb- 2022].

[10] "Dynamic Host Configuration Protocol - Wikipedia", *En.wikipedia.org*, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol. [Accessed: 19- Feb- 2022].