



FINAL%

FACULTY OF SCIENCE AGRICULTURE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE
ASSIGNMENT COVER SHEET

| | |
|------------------|-----------------------------|
| MODULE TITLE | Wireless Networks |
| MODULE CODE | 4CPS504 |
| ASSIGNMENT TOPIC | Exploring Wi-Fi Raw Packets |
| LECTURER NAME | Prof A Terzoli |
| DUE DATE | 9 May 2025 |

NON - PLAGIARISM DECLARATION

I know that plagiarism means taking and using the ideas, writings, works or inventions of another as if they were one's own. I know that plagiarism not only includes verbatim copying, but also the extensive use of another person's ideas without proper acknowledgement (which includes the proper use of quotation marks). I know that plagiarism covers this sort of use of material found in textual sources and from the Internet. I acknowledge and understand that plagiarism is wrong. I understand that my research must be accurately referenced. I have followed the rules and conventions concerning referencing, citation and the use of quotations as set out in the Departmental Guide. This assignment is my own work, or my group's own unique group assignment. I acknowledge that copying someone else's assignment, or part of it, is wrong, and that submitting identical work to others constitutes a form of plagiarism. I have not allowed, nor will I in the future allow anyone to copy my work with the intention of passing it off as their own work. By signing this cover sheet, I agree that I have read and understood the above. I acknowledge that should it be found to be higher than the acceptable similarity percentage, I may receive 0 (ZERO) for my assignment.

| STUDENT NAME | STUDENT NO | SIGNATURE |
|---------------------|------------|-----------|
| Sphamandla Njokweni | 202126142 | |
| LECTURER REMARKS | | |
| | | |

Question 1

The Devices :

The two machines was used in creating or generating the capture files:

- **Linux Machine (desktop)** : The Linux system with a Wi-Fi network interface working in

monitor mode

- **Windows Machine (Laptop)** : A Windows 10 system with a Wi-Fi network interface working in managed mode.

Wi-Fi Network :

The Linux and window devices was operating or working within the same WiFi network or connected to the same WiFi network which was set up using Access points (AP), The access point functioned as the central hub in an infrastructure mode WiFi network, permitting the wireless communication between devices and providing internet or LAN access

Windows computer : The windows computer's managed mode network interface was used to record the Wi-Fi frames. Wireshark which was installed on the windows computer made the capture process easier.

Capture files

The capture files "connectingAndDisconnecting-Dlink-fakeEthernetFrames-windows10Managed.pcapng" was created on the windows computer and includes Wi-Fi frames that have been contained within Ethernet frames. The raw Wi-Fi frames collected by the network interface running in monitor mode are contained in the "connectingAndDisconnecting-Dlink-WifiFrames-LinuxMonitor.pcapng" capture file, which was created on the Linux computer

Capture method :

Linux machine : The Linux machine's network interface, running in a monitored was used to record the Wi-Fi frames. In a similar manner, packets have been captured on Linux systems using wireshark

Question 2

"connectingAndDisconnecting-Dlink-FakeEthernetFrames-Windows10Managed.pcapng":

>This capture file was created on a Windows 10 controlled system and seems to involve a D-Link device

>The Wi-Fi frames that were captured and included in this file are displayed as fake Ethernet frames and this implies that they were encased within Ethernet frames for testing or compatibility reasons.

>Examining how the Windows 10 management system manages Wi-Fi connections and disconnections with D-link equipment is the main goal of this capture file.

> The Wi-Fi interface mode is the managed mode

>it only displays or show Ethernet frames at the data link layer which are derived from higher-layer protocols like as TCP, DHCP, IP and so on.

> it fails to record real 802.11 Wi-Fi frames

"connectingAndDisconnecting-Dlink-WifiFrames-linuxMonitor.pcapng":

>This capture file was created on Linux system running in a monitor mode but it is also included a D-link device

>The capture file offers more through details on Wi-Fi specific parameters by containing raw Wi-Fi frames that were recorded by the network card when it was in monitored mode.

>The Wi-Fi frames that were captured and included in this file are displayed as original Ethernet frames

>The Wi-Fi interface mode is the monitor mode

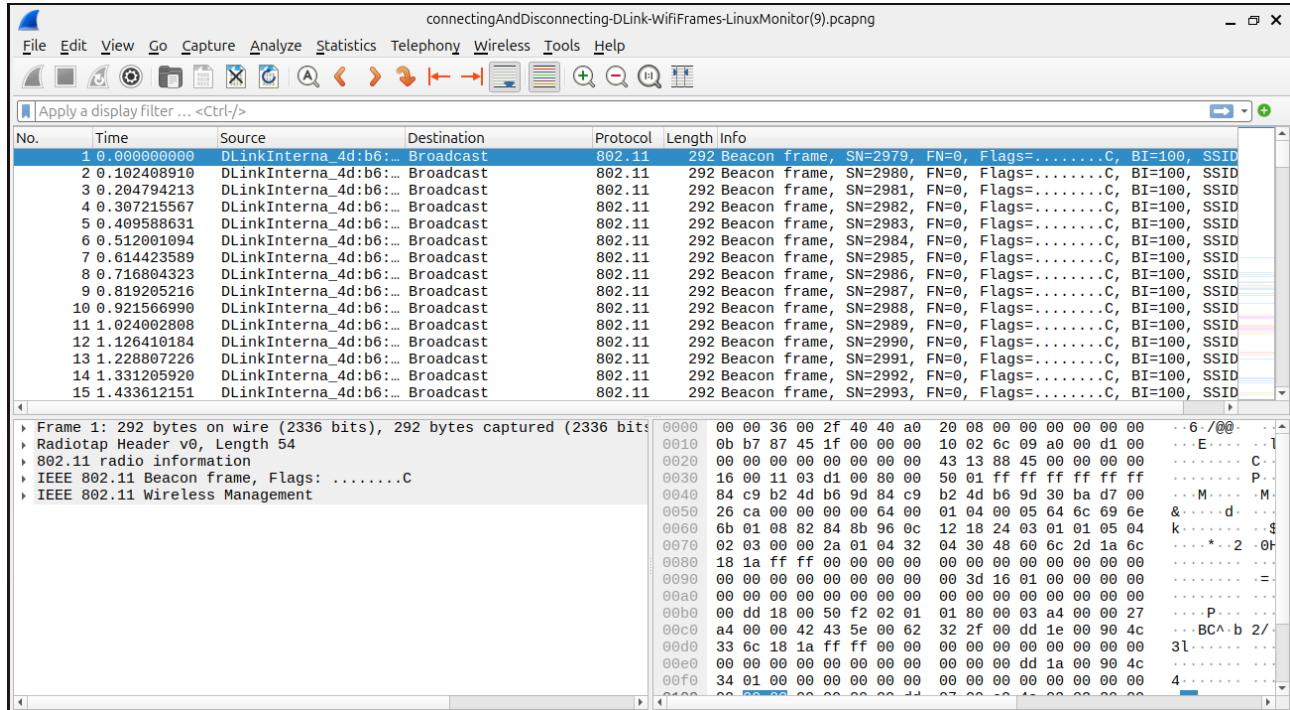
> it records unprocessed or raw 802.11 Wi-Fi frames such as the Beacon frames, retransmissions and many more.

Question 3

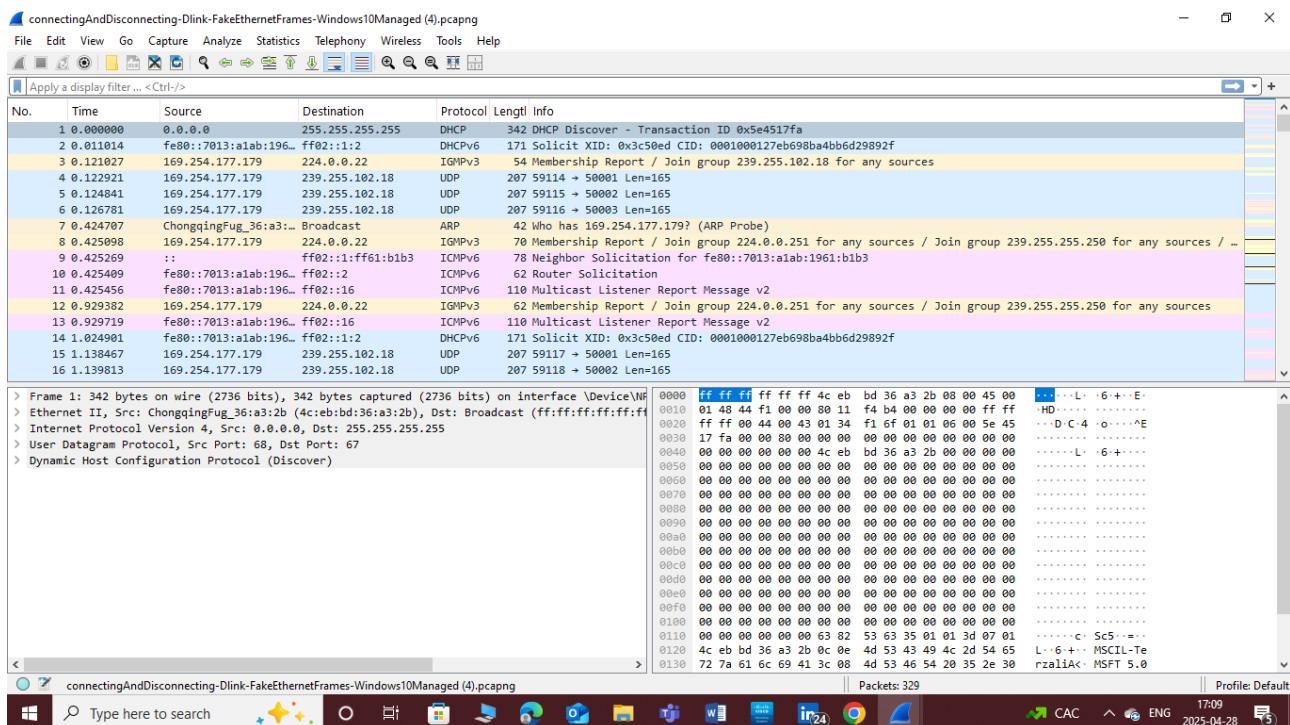
Aligning the packets in both capture files according to the same events is the main aim of synchronization. To put it another way, we must locate a packet that corresponds to the same event and shows up in both capture files. After locating such a packet, we may synchronize the two capture files by using its timestamps to ascertain the time shift between them.

step-by-step approach that I followed to synchronize the capture files :

"connectingAndDisconnecting-Dlink-WifiFrames-LinuxMonitor.pcapng"

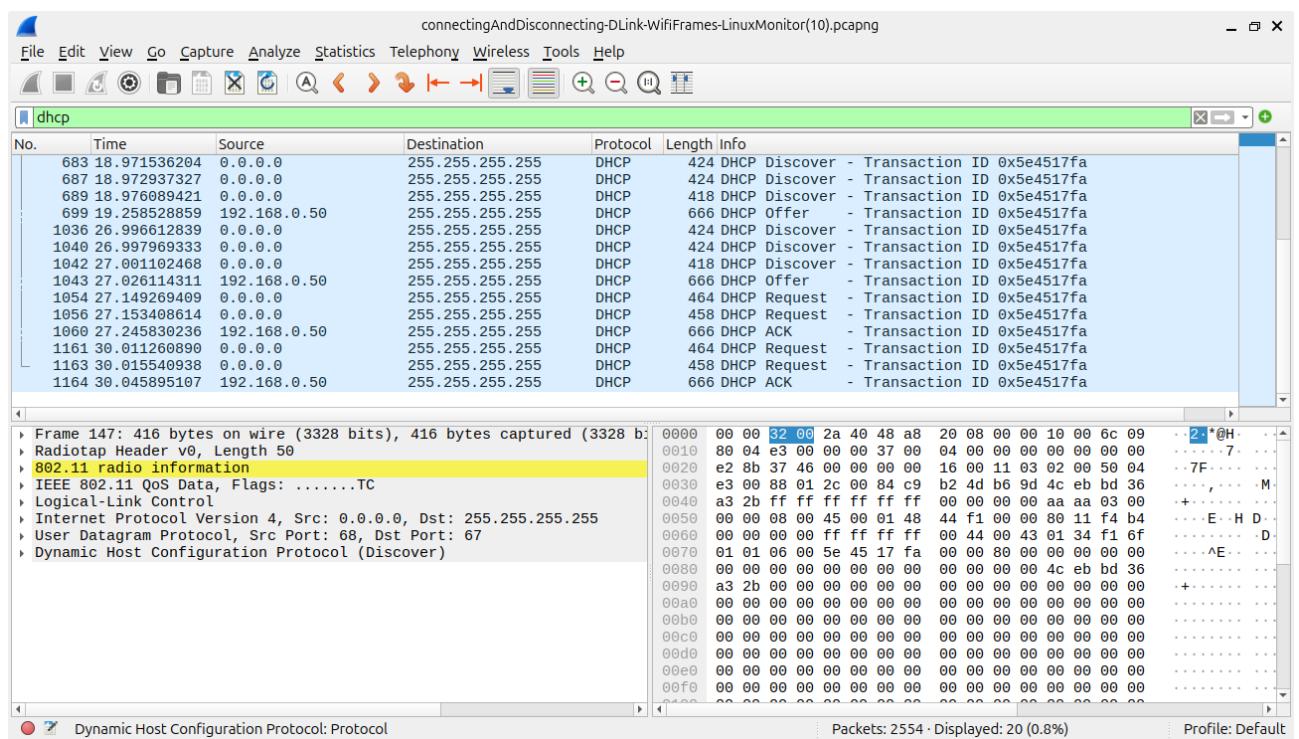


"connectingAndDisconnecting-Dlink-FakeEthernetFrames-Windows10Managed.pcapng"

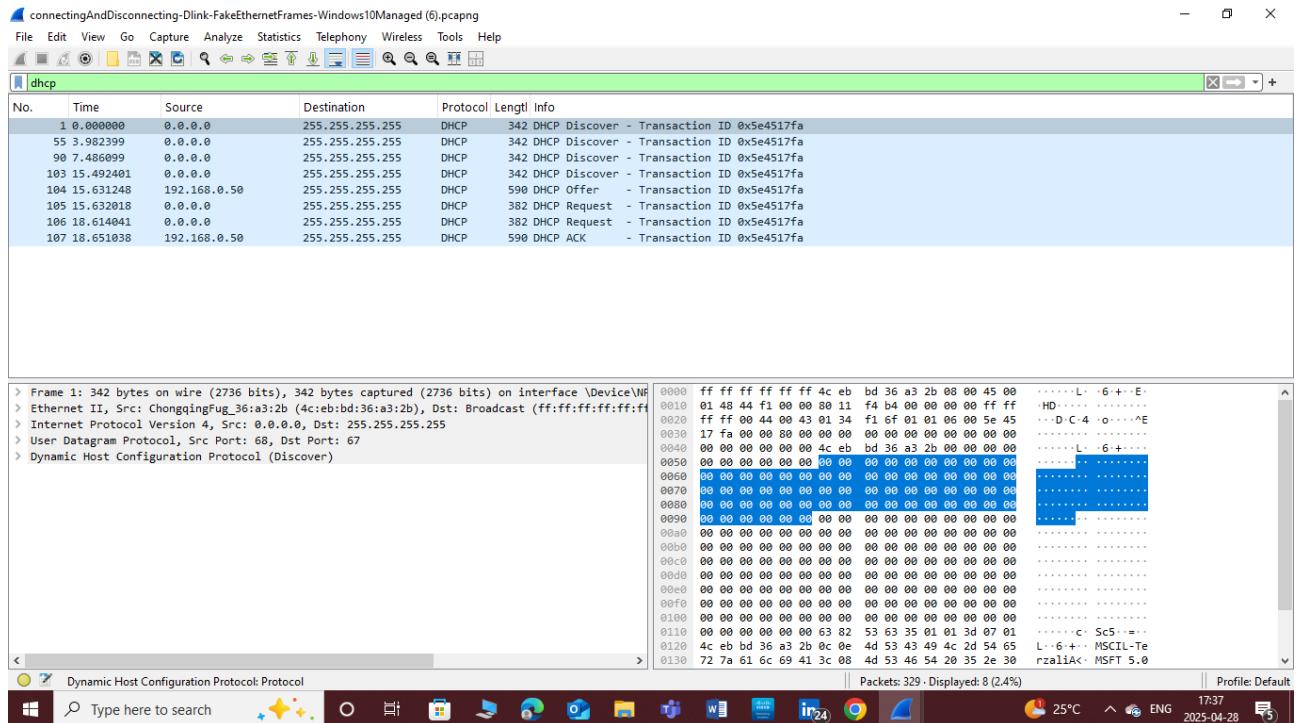


I applied the display filter on both files to display only the ICMP packets.

For Linux machine :



for window machine



> I identified the packet that appears in both capture files and represents the same event. This can serve as a reference point for synchronization.

"connectingAndDisconnecting-Dlink-WifiFrames-LinuxMonitor.pcapng"



"connectingAndDisconnecting-Dlink-FakeEthernetFrames-Windows10Managed.pcapng"



The aforementioned packets can be regarded as representing the same event because they are part of the same protocol, their source and destination addresses match, and they contain the same transaction.

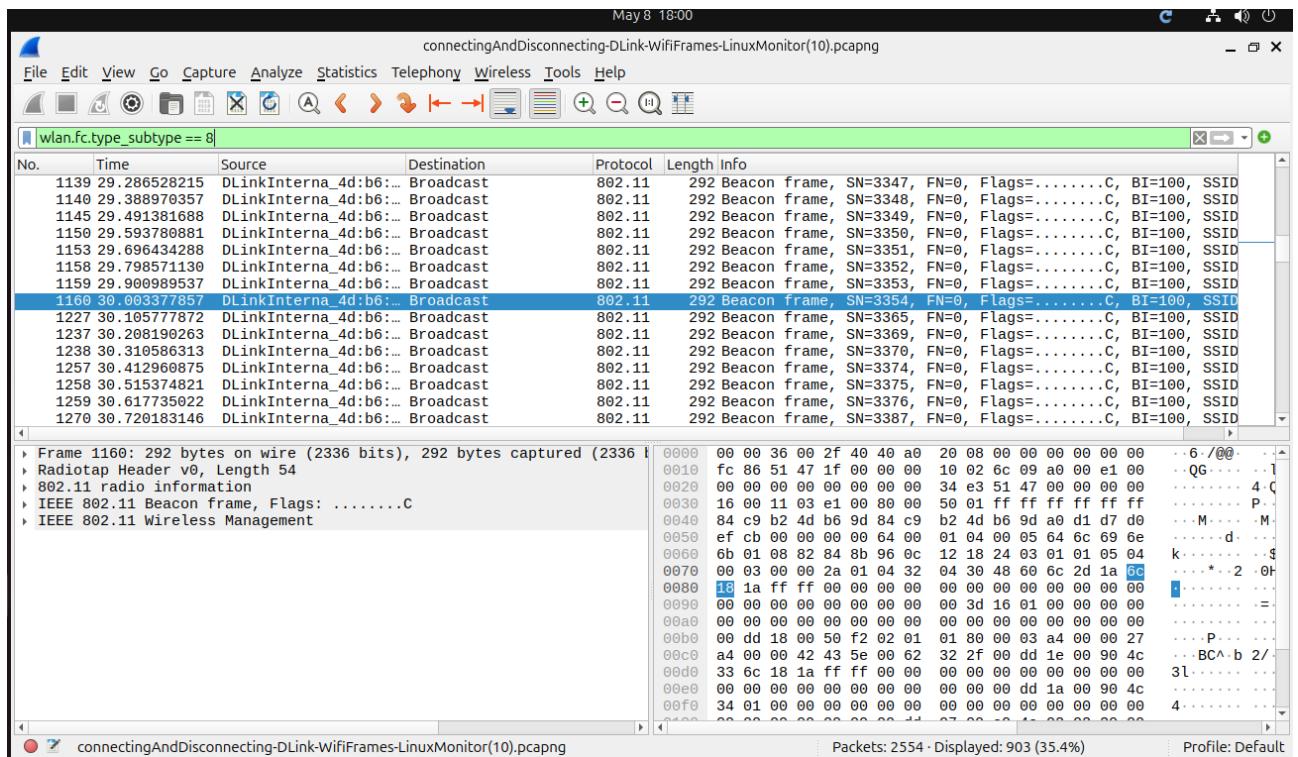
We take note of the timestamp of the common ICMP packet in each capture file.
In the file with fake Ethernet frame, the timestamp is 11.3958,

for linux machine : time = 1164 30.045895107

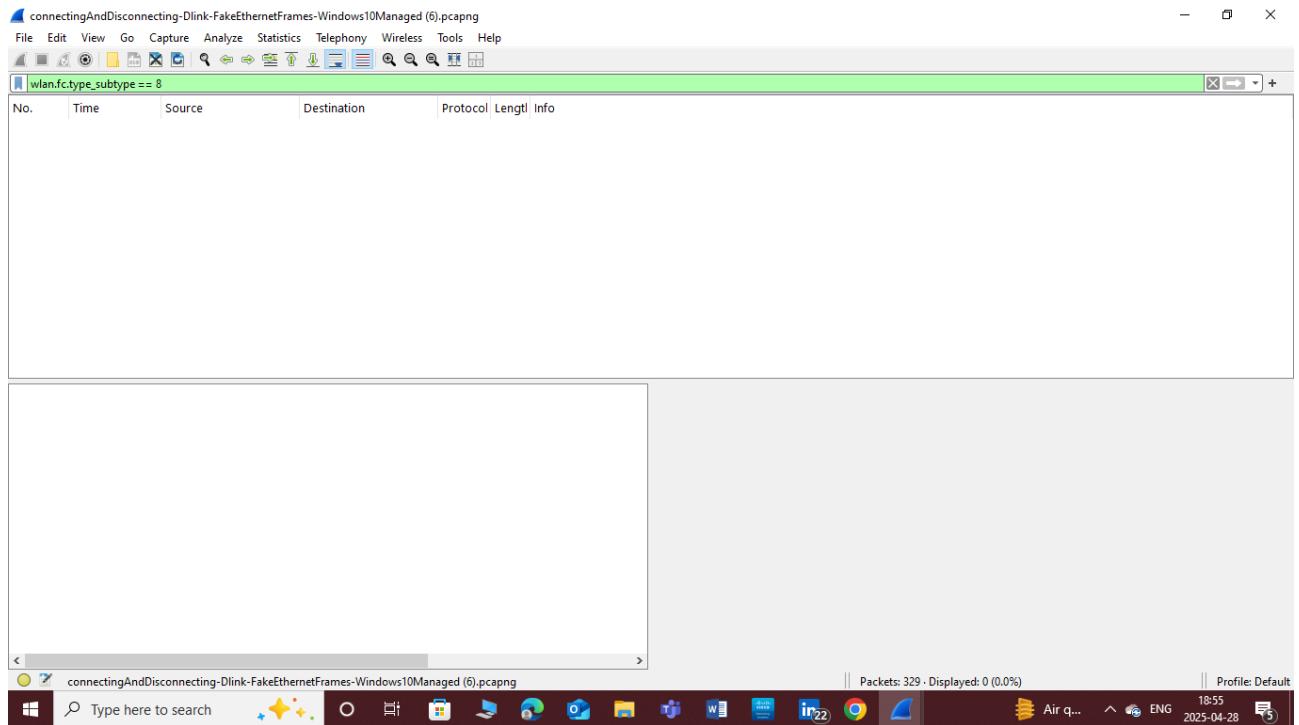
for window machine : time = 107 18.651038

Question 4

I opened the two files in Wireshark which are "connectingAndDisconnecting-Dlink-FakeEthernetFrames-Windows10Managed.pcapng" and "connectingAndDisconnecting-Dlink-WifiFrames-LinuxMonitor.pcapng". Then I applied a display filter to show only beacon packets. I used the display filter 'wlan.fc.type_subtype == 8' to filter and display only beacon frames.



"connectingAndDisconnecting-Dlink-FakeEthernetFrames-Windows10Managed.pcapng"



Because the Windows computer is a fake Ethernet frame, the beacon frames are only present for Wi-Fi frames that are running on Linux computer or machine.

In the Linux capture file, the beacon frames' sequence numbers gradually increases by 1.

By detecting beacon packets, examining their sequence numbers, and measuring the time gaps between them, we can gain insights into how the access point (AP) transmits beacons. This analysis helps us better understand the network's functioning and the timing and order in which the AP sends beacon frames.

question 5

Association Dialog :

The **Association Dialog** is a crucial part of how a wireless client connects to a Wi-Fi network. After detecting a network either through beacon frames sent by the access point (AP) or by actively probing for available networks the client first performs an authentication step, which is often a simple handshake in open networks. Once authenticated, the client initiates the association process by sending an **Association Request** frame to the AP. This request includes information such as the network name (SSID), supported data rates, and the client's capabilities. The AP processes this request and replies with an **Association Response** frame, which either accepts or rejects the request. If accepted, the response includes an association ID and confirms the parameters for the connection. Once this exchange is complete, the client is considered associated with the AP and can proceed to obtain an IP address and start data communication. This dialog ensures that both the client and AP agree on connection settings before allowing network access

Disassociation Dialog :

The **Disassociation Dialog** is the formal process by which a wireless client or an access point (AP) terminates an established connection in a Wi-Fi network. This communication occurs through the exchange of **Disassociation frames**. When a client decides to leave the network such as when the user turns off Wi-Fi, moves out of range, or switches to another network it sends a Disassociation frame to the AP to inform it that the session is ending. Conversely, the AP can also initiate disassociation if, for example, the client violates network policies, experiences signal issues, or the AP is shutting down. The Disassociation frame includes a reason code explaining why the connection is being terminated. This process allows both the AP and the client to gracefully release network resources, update their connection tables, and prepare for either reconnection or a clean disconnection. It helps maintain stability and proper resource management in the wireless network.

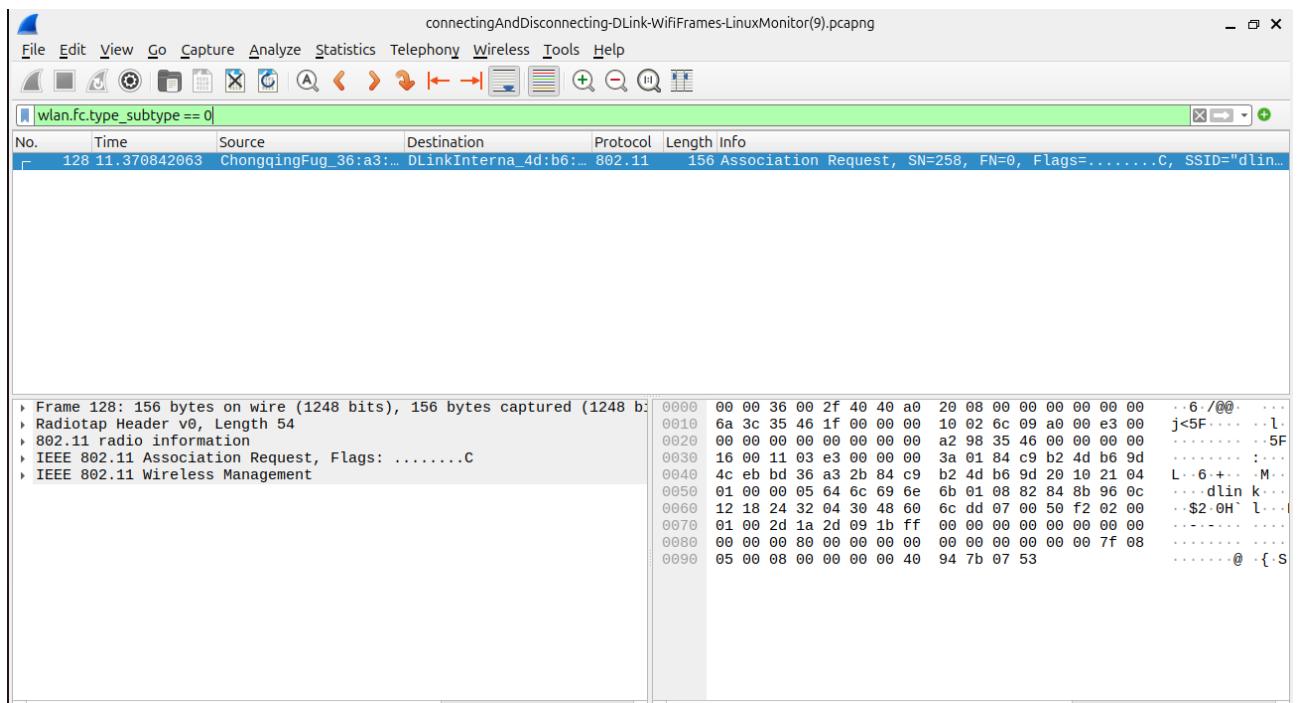
Steps to find the association and disassociation dialog in the two given files, I started by opening the two files on wireshark

>Both capture file which is "connectingAndDisconnecting-Dlink-WifiFrames-LinuxMonitor.pcapng" and "connectingAndDisconnecting-

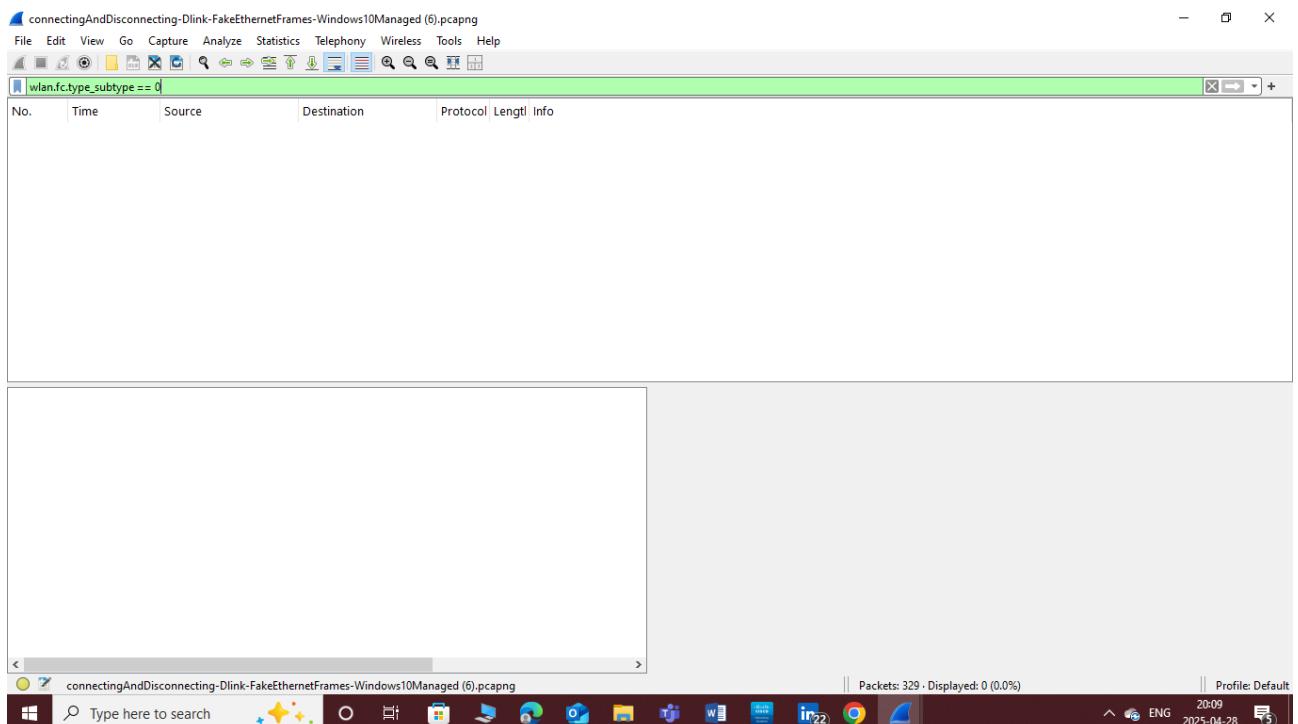
"Dlink-FakeEthernetFrames-Windows10Managed.pcapng" was opened in wireshark. I applied the display filter to show only association and disassociation packets. I then applied the filter `wlan.fc.type_subtype == 0` to filter and only display association and disassociation packets.

The screen short are as below for both Linux and windows machines :

For Linux machine :



for windows machine

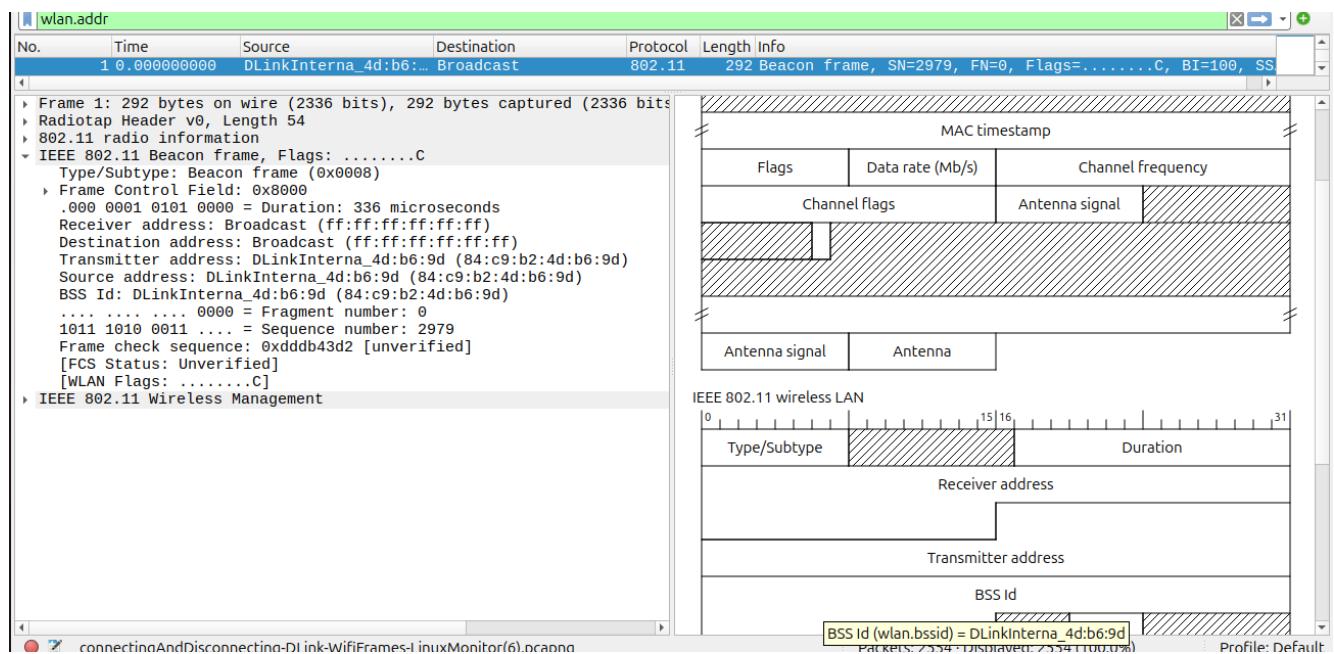


The two screen short above shows the two capture files that was opened on wireshark after applying the display filter to show the association and disassociation packets. And it can be observed that the association and disassociation packets only present on

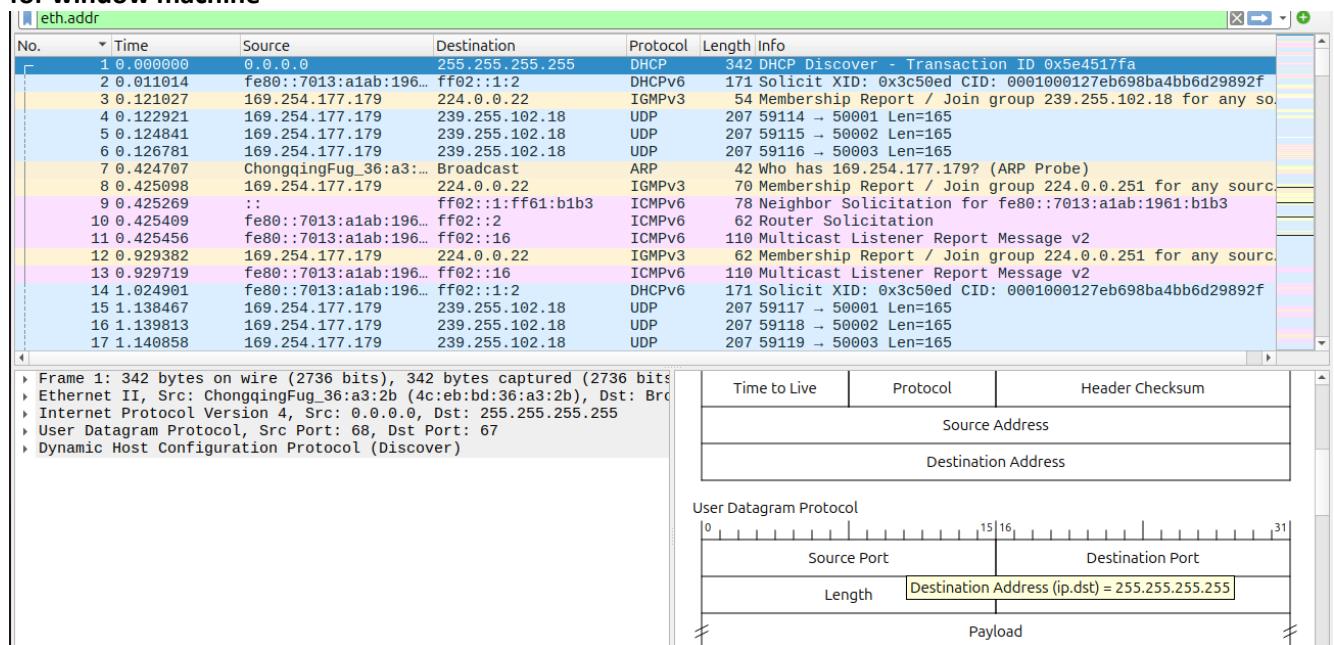
the "connectingAndDisconnecting-DLink-WifiFrames-LinuxMonitor.pcapng" capture file only, it is not present on the windows machine as proven in the above screen short.

Question 6

screen short for Linux machine



for window machine

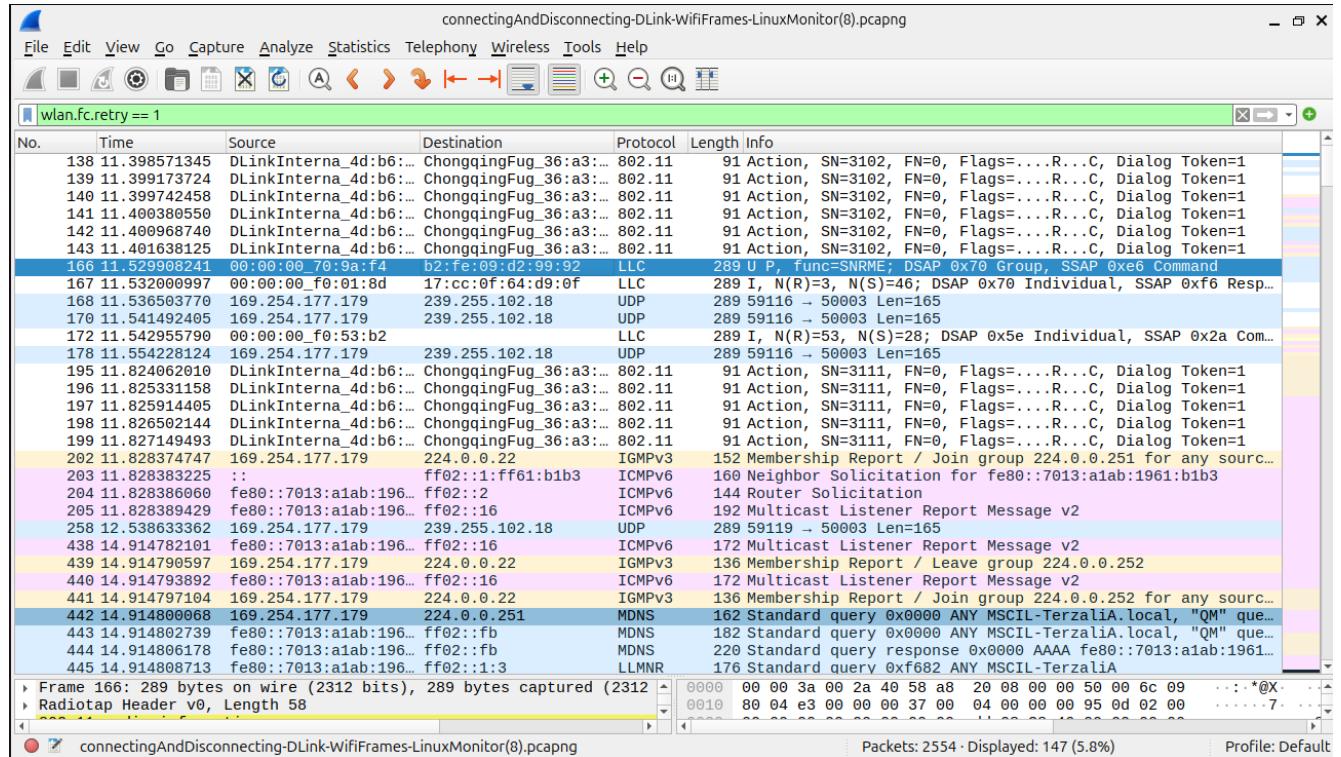


In the first packet captured on the Windows machine (in Managed mode), only two MAC addresses are present because the wireless network interface presents the data in a simplified, Ethernet-like format. In this mode, the network adapter strips away the 802.11-specific header information, leaving only the source and destination MAC addresses. This is sufficient for normal Ethernet-style communication because the access point (AP) handles the wireless routing and acts as a transparent bridge between the client and the broader network. As a result, there's no need to explicitly include the BSSID (the MAC address of the AP) in each packet the AP is assumed to be part of the infrastructure behind the scenes.

In contrast, the second packet captured on the Linux machine (in Monitor mode) includes three MAC addresses because it retains the

full 802.11 frame structure. In Wi-Fi infrastructure mode, these three addresses are essential to properly route the frame through the wireless network. The frame must include the **transmitter address** (the device physically sending the frame), the **receiver address** (the intended recipient), and the **BSSID** (the identifier of the access point managing the connection). This extra address helps distinguish between the source of the data and the network infrastructure relaying it a crucial detail in wireless communication where multiple devices may share the same medium through a central AP. Thus, three addresses are required in Monitor mode to fully represent the structure and routing context of the wireless frame.

Question 7



In the provided screenshot from the Wireshark capture file `connectingAndDisconnecting-DLink-WifiFrames-LinuxMonitor(8).pcapng`, we can observe a sequence of retransmissions that is indicative of typical wireless communication behavior when acknowledgments (ACKs) are delayed or lost.

Focusing on frame 166, we see that the frame originates from `00:00:00:70:9a:f4` and is being transmitted to `b2:fe:09:d2:99:92`, using the LLC protocol. It has a length of 289 bytes and includes a field marked `wlan.fc.retry = 1`, which is a clear indicator that this is a retransmission of an earlier frame that did not receive an acknowledgment. This retry bit being set suggests that the same data frame has been sent before and is now being sent again because the sender did not receive an ACK from the receiver. Following this, the surrounding frames (such as frames 167 and beyond) do not appear to be direct ACKs (they are different in protocol or source/destination), indicating that the retransmission sequence may still be ongoing at that point in the capture. The retransmission sequence will eventually end when an ACK is received.

Question 8

The data packets appear to be unencrypted. When viewing the packet contents in Wireshark, the information is clearly visible and lacks the typical features of encrypted data, such as random or unreadable patterns. This strongly suggests that the transmission is in plaintext. Moreover, examining the frame control field reveals that the "Protected Frame bit" or "protected flag" which usually indicates encryption is not active. This absence further supports the conclusion that the data has not been encrypted. Therefore, based on the clear readability of the payload and the lack of encryption markers, it is evident that the packets were sent without encryption.