

Name: Seif Eldin Mahmoud Ibrahim, ID: 6773

Lab 1 Report, Group 1, Section 2

- Arranging in ascending "A"
Client Response

The screenshot displays a network traffic capture in Wireshark. The left pane shows the packet list with a filter of 'tcp.port == 5050'. The right pane shows the packet details for the selected packet (No. 7), including the Ethernet II, Internet Protocol, and Transmission Control Protocol (TCP) layers. The TCP layer shows a sequence number of 56 and a length of 12 bytes. The packet data is shown in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
5	14.106831	192.168.56.1	192.168.56.1	TCP	108	50251 → 5050 [PS...]
6	14.106864	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [AC...]
7	14.106881	192.168.56.1	192.168.56.1	TCP	56	50251 → 5050 [PS...]
8	14.106896	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [AC...]
9	14.107111	192.168.56.1	192.168.56.1	TCP	55	5050 → 50251 [PS...]
10	14.107126	192.168.56.1	192.168.56.1	TCP	44	50251 → 5050 [AC...]

Frame 7: 56 byte
Null/Loopback
Internet Protocol
Transmission Control Protocol
Data (12 bytes)

Server Response

The screenshot displays a network traffic capture in Wireshark. The left pane shows the packet list with a filter of 'tcp.port == 5050'. The right pane shows the packet details for the selected packet (No. 9), including the Ethernet II, Internet Protocol, and Transmission Control Protocol (TCP) layers. The TCP layer shows a sequence number of 55 and a length of 11 bytes. The packet data is shown in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
5	14.106831	192.168.56.1	192.168.56.1	TCP	108	50251 → 5050 [PS...]
6	14.106864	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [AC...]
7	14.106881	192.168.56.1	192.168.56.1	TCP	56	50251 → 5050 [PS...]
8	14.106896	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [AC...]
9	14.107111	192.168.56.1	192.168.56.1	TCP	55	5050 → 50251 [PS...]
10	14.107126	192.168.56.1	192.168.56.1	TCP	44	50251 → 5050 [AC...]

Frame 9: 55 byte
Null/Loopback
Internet Protocol
Transmission Control Protocol
Data (11 bytes)

- Arrange in descending “D”
Client Response

The screenshot displays a Python client application running in a terminal window and a Wireshark network traffic capture. The client application, titled "Lab1 Network", shows a menu with options: "1- Send Message to server" and "2- Disconnect from server". The user has chosen "1" and entered the string "Dhbnmksbjhk". The terminal output shows the server's response: "Ajnjdscnjhs".

The Wireshark capture, titled "Adapter for loopback traffic capture", shows a list of network packets. The filter is set to "tcp.port == 5050". The packet list shows several TCP packets from 192.168.56.1 to 192.168.56.1. The packet details pane shows the selected packet (No. 478) with a length of 55 bytes, containing a Null/Loopback, Internet Protocol, and Transmission Control Protocol (TCP) header. The packet data (11 bytes) is shown in hexadecimal and ASCII.

Server Response

The screenshot displays a Python server application running in a terminal window and a Wireshark network traffic capture. The server application, titled "Lab1 Network", shows a menu with options: "1- Send Message to server" and "2- Disconnect from server". The user has chosen "1" and entered the string "Dhbnmksbjhk". The terminal output shows the server's response: "Ajnjdscnjhs".

The Wireshark capture, titled "Adapter for loopback traffic capture", shows a list of network packets. The filter is set to "tcp.port == 5050". The packet list shows several TCP packets from 192.168.56.1 to 192.168.56.1. The packet details pane shows the selected packet (No. 480) with a length of 54 bytes, containing a Null/Loopback, Internet Protocol, and Transmission Control Protocol (TCP) header. The packet data (10 bytes) is shown in hexadecimal and ASCII.

• Capitalization “C” Client Response

The screenshot displays a Python client application running in a terminal window and a Wireshark network traffic capture. The client application, named 'Client.py', is running on a Windows system. The terminal output shows the client's choice to send a message to the server, followed by the string 'Cadsjbnmklkjh'. The Wireshark capture shows a TCP packet from the client to the server, with the payload 'Cadsjbnmklkjh'.

Client.py Output:

```
snmkjhhbb
Enter your choice:
1- Send Message to server
2- Disconnect from server
Your Choice: 1
Enter your String: Cadsjbnmklkjh
ADSJBNMKLKJH
Enter your choice:
1- Send Message to server
2- Disconnect from server
Your Choice: 
```

Wireshark Capture:

No.	Time	Source	Destination	Protocol	Length	Info
7	14.106881	192.168.56.1	192.168.56.1	TCP	56	50251 → 5050 [...]
8	14.106896	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [...]
9	14.107111	192.168.56.1	192.168.56.1	TCP	55	5050 → 50251 [...]
10	14.107126	192.168.56.1	192.168.56.1	TCP	44	50251 → 5050 [...]
476	1709.965160	192.168.56.1	192.168.56.1	TCP	108	50251 → 5050 [...]
477	1709.965258	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [...]
478	1709.965371	192.168.56.1	192.168.56.1	TCP	55	50251 → 5050 [...]
479	1709.965395	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [...]
480	1709.965839	192.168.56.1	192.168.56.1	TCP	54	5050 → 50251 [...]
481	1709.965861	192.168.56.1	192.168.56.1	TCP	44	50251 → 5050 [...]
534	1945.029096	192.168.56.1	192.168.56.1	TCP	108	50251 → 5050 [...]
535	1945.029136	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [...]
536	1945.029167	192.168.56.1	192.168.56.1	TCP	57	50251 → 5050 [...]
537	1945.029178	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [...]
538	1945.029445	192.168.56.1	192.168.56.1	TCP	56	5050 → 50251 [...]
539	1945.029472	192.168.56.1	192.168.56.1	TCP	44	50251 → 5050 [...]

Frame 536: 57 by

- 0000 02 00 00 00 45 00 00 35 65 5c 40 00 00 06 00 00 ...E..S e@.....
- 0010 c0 a8 38 01 c0 a8 38 01 c4 4b 13 ba ee 2a 4e 56 ...8...8...K...NV
- 0020 01 35 b0 9e 50 18 04 ff 33 97 00 00 43 61 64 73 ...S-P...3...Cads
- 0030 6a 62 6e 6d 6b 6c 6b 6a 68 jbnmklkj h

Data (13 bytes)

Server Response

The screenshot displays a Python server application running in a terminal window and a Wireshark network traffic capture. The server application, named 'Server.py', is running on a Windows system. The terminal output shows the server listening on port 50251, receiving a connection from 192.168.56.1, and receiving the string 'Cadsjbnmklkjh'. The Wireshark capture shows a TCP packet from the server to the client, with the payload 'BNMKLKJH'.

Server.py Output:

```
snmkjhhbb
Enter your choice:
1- Send Message to server
2- Disconnect from server
Your Choice: 1
Enter your String: Cadsjbnmklkjh
ADSJBNMKLKJH
Enter your choice:
1- Send Message to server
2- Disconnect from server
Your Choice: 
```

Wireshark Capture:

No.	Time	Source	Destination	Protocol	Length	Info
7	14.106881	192.168.56.1	192.168.56.1	TCP	56	50251 → 5050 [...]
8	14.106896	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [...]
9	14.107111	192.168.56.1	192.168.56.1	TCP	55	5050 → 50251 [...]
10	14.107126	192.168.56.1	192.168.56.1	TCP	44	50251 → 5050 [...]
476	1709.965160	192.168.56.1	192.168.56.1	TCP	108	50251 → 5050 [...]
477	1709.965258	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [...]
478	1709.965371	192.168.56.1	192.168.56.1	TCP	55	50251 → 5050 [...]
479	1709.965395	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [...]
480	1709.965839	192.168.56.1	192.168.56.1	TCP	54	5050 → 50251 [...]
481	1709.965861	192.168.56.1	192.168.56.1	TCP	44	50251 → 5050 [...]
534	1945.029096	192.168.56.1	192.168.56.1	TCP	108	50251 → 5050 [...]
535	1945.029136	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [...]
536	1945.029167	192.168.56.1	192.168.56.1	TCP	57	50251 → 5050 [...]
537	1945.029178	192.168.56.1	192.168.56.1	TCP	44	5050 → 50251 [...]
538	1945.029445	192.168.56.1	192.168.56.1	TCP	56	5050 → 50251 [...]
539	1945.029472	192.168.56.1	192.168.56.1	TCP	44	50251 → 5050 [...]

Frame 538: 56 by

- 0000 02 00 00 00 45 00 00 34 65 5e 40 00 00 06 00 00 ...E..4 e@.....
- 0010 c0 a8 38 01 c0 a8 38 01 13 ba c4 4b 01 35 b0 9e ...8...8...K...S
- 0020 ee 2a 4e 63 50 18 20 f9 1c 51 00 00 41 44 53 4a ...*NcP...Q...ADSJ
- 0030 42 4e 4d 4b 4c 4b 4a 48 BNMKLKJH

Data (12 bytes)

- If any alphabet neither A, C, nor D
- ## Client Response

The screenshot displays a Python client application running in a terminal window and a Wireshark network traffic capture. The client application is a simple server-client interaction. The user enters a string 'Kjhjbhbjnkm1' and chooses to send a message to the server. The server responds with the same string. The Wireshark capture shows the network traffic between the client and the server. The packet list shows a TCP connection established between 192.168.56.1 and 192.168.56.1. The packet details show the data being sent and received. The packet bytes show the raw data being transmitted.

Client Application Output:

```
PS D:\My Projects\Python\Lab1 Network> python Client.py
Enter your choice:
1- Send Message to server
2- Disconnect from server
Your Choice: 1
Enter your String: Kjhjbhbjnkm1

Kjhjbhbjnkm1

Enter your choice:
1- Send Message to server
2- Disconnect from server
Your Choice: 1
```

Wireshark Packet Capture:

No.	Time	Source	Destination	Protocol	Length	Info
883	2269.509857	192.168.56.1	192.168.56.1	TCP	44	5050 → 51652 [..
884	2269.510065	192.168.56.1	192.168.56.1	TCP	54	5050 → 51652 [..
885	2269.510083	192.168.56.1	192.168.56.1	TCP	44	51652 → 5050 [..
886	2269.510113	192.168.56.1	192.168.56.1	TCP	44	5050 → 51652 [..
887	2269.510126	192.168.56.1	192.168.56.1	TCP	44	51652 → 5050 [..
888	2269.513195	192.168.56.1	192.168.56.1	TCP	44	51652 → 5050 [..
889	2269.513260	192.168.56.1	192.168.56.1	TCP	44	5050 → 51652 [..
902	2286.483643	192.168.56.1	192.168.56.1	TCP	56	51664 → 5050 [..
903	2286.483702	192.168.56.1	192.168.56.1	TCP	56	5050 → 51664 [..
904	2286.483751	192.168.56.1	192.168.56.1	TCP	44	51664 → 5050 [..
907	2292.519594	192.168.56.1	192.168.56.1	TCP	108	51664 → 5050 [..
908	2292.519632	192.168.56.1	192.168.56.1	TCP	44	5050 → 51664 [..
909	2292.519648	192.168.56.1	192.168.56.1	TCP	57	51664 → 5050 [..
910	2292.519656	192.168.56.1	192.168.56.1	TCP	44	5050 → 51664 [..
911	2292.519901	192.168.56.1	192.168.56.1	TCP	57	5050 → 51664 [..
912	2292.519923	192.168.56.1	192.168.56.1	TCP	44	51664 → 5050 [..

Packet Details (Frame 909):

- Frame 909: 57 by 0000 02 00 00 00 45 00 00 35 65 88 40 00 80 06 00 00 ...E: 5 e@....
- Null/Loopback
- Internet Protocol Version 4
- Transmission Control Protocol
- Data (13 bytes)

Packet Bytes:

```
0000 02 00 00 00 45 00 00 35 65 88 40 00 80 06 00 00 ...E: 5 e@....
0010 c0 a8 38 01 c0 a8 38 01 c9 d0 13 ba ec e8 75 b1 ...8: 8: .....u.
0020 d2 ef d4 89 50 18 20 fa e7 57 00 00 4b 6a 68 6a ...P: 7: 3: Kjhj
0030 62 68 76 62 6a 6e 6b 6d 6c                      bhbjnkm 1
```

Server Response

The screenshot displays a Python server application running in a terminal window and a Wireshark network traffic capture. The server application is a simple server-client interaction. The user enters a string 'Kjhjbhbjnkm1' and chooses to send a message to the server. The server responds with the same string. The Wireshark capture shows the network traffic between the client and the server. The packet list shows a TCP connection established between 192.168.56.1 and 192.168.56.1. The packet details show the data being sent and received. The packet bytes show the raw data being transmitted.

Client Application Output:

```
PS D:\My Projects\Python\Lab1 Network> python Client.py
Enter your choice:
1- Send Message to server
2- Disconnect from server
Your Choice: 1
Enter your String: Kjhjbhbjnkm1

Kjhjbhbjnkm1

Enter your choice:
1- Send Message to server
2- Disconnect from server
Your Choice: 1
```

Wireshark Packet Capture:

No.	Time	Source	Destination	Protocol	Length	Info
883	2269.509857	192.168.56.1	192.168.56.1	TCP	44	5050 → 51652 [..
884	2269.510065	192.168.56.1	192.168.56.1	TCP	54	5050 → 51652 [..
885	2269.510083	192.168.56.1	192.168.56.1	TCP	44	51652 → 5050 [..
886	2269.510113	192.168.56.1	192.168.56.1	TCP	44	5050 → 51652 [..
887	2269.510126	192.168.56.1	192.168.56.1	TCP	44	51652 → 5050 [..
888	2269.513195	192.168.56.1	192.168.56.1	TCP	44	51652 → 5050 [..
889	2269.513260	192.168.56.1	192.168.56.1	TCP	44	5050 → 51652 [..
902	2286.483643	192.168.56.1	192.168.56.1	TCP	56	51664 → 5050 [..
903	2286.483702	192.168.56.1	192.168.56.1	TCP	56	5050 → 51664 [..
904	2286.483751	192.168.56.1	192.168.56.1	TCP	44	51664 → 5050 [..
907	2292.519594	192.168.56.1	192.168.56.1	TCP	108	51664 → 5050 [..
908	2292.519632	192.168.56.1	192.168.56.1	TCP	44	5050 → 51664 [..
909	2292.519648	192.168.56.1	192.168.56.1	TCP	57	51664 → 5050 [..
910	2292.519656	192.168.56.1	192.168.56.1	TCP	44	5050 → 51664 [..
911	2292.519901	192.168.56.1	192.168.56.1	TCP	57	5050 → 51664 [..
912	2292.519923	192.168.56.1	192.168.56.1	TCP	44	51664 → 5050 [..

Packet Details (Frame 911):

- Frame 911: 57 by 0000 02 00 00 00 45 00 00 35 65 8a 40 00 80 06 00 00 ...E: 5 e@....
- Null/Loopback
- Internet Protocol Version 4
- Transmission Control Protocol
- Data (13 bytes)

Packet Bytes:

```
0000 02 00 00 00 45 00 00 35 65 8a 40 00 80 06 00 00 ...E: 5 e@....
0010 c0 a8 38 01 c0 a8 38 01 13 ba c9 d0 d2 ef d4 89 ...8: 8: .....u.
0020 ec e8 75 be 50 18 20 fa e7 4a 00 00 4b 6a 68 6a ...u: P: 7: 3: Kjhj
0030 62 68 76 62 6a 6e 6b 6d 6c                      bhbjnkm 1
```