Name- Aman Panda
CCE - B
210953038
Roll No - 06

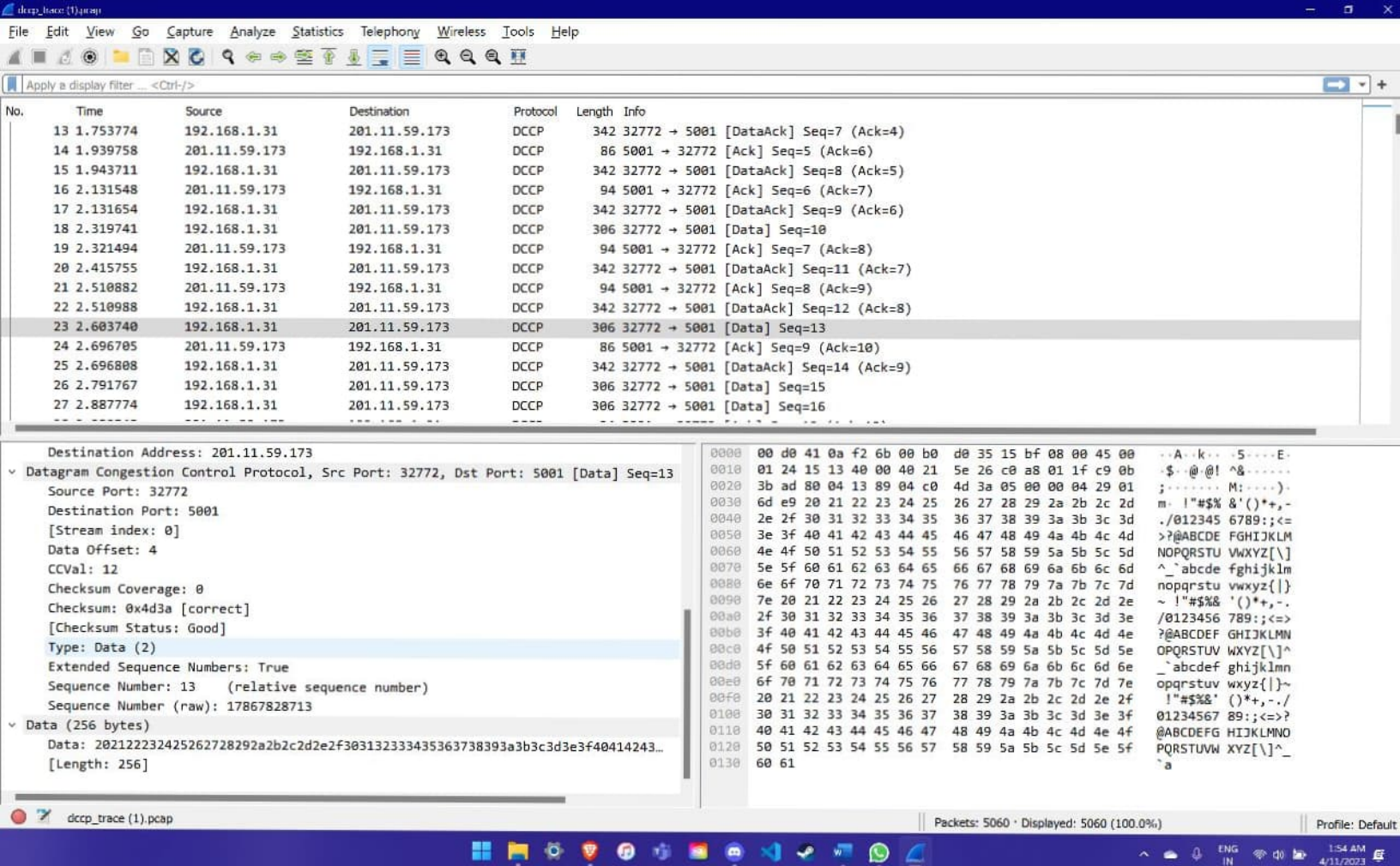
CNP FISAC-1

Group- DCCP
ICT- 2255

__/__/____

(b) To generate and capture DCCP packets using wireshark, follow these steps:

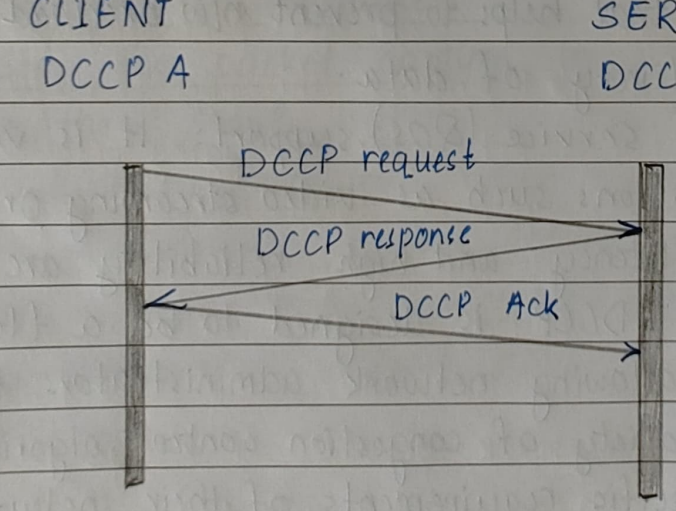
1. Install and open wireshark on your computer.
2. Select the network interface that you want to capture DCCP packets from.
3. Click on the "capture options" button in the main menu to open the capture options dialog box.
4. In the "Capture Filter" field, enter "dccp" to filter out all non-DCCP packets.
5. Click on the "start" button to begin the packet capture.
6. Start the application that will generate the DCCP traffic you want to capture.
7. Perform the actions in the application that will generate DCCP packets.
8. Stop the packet capture in Wireshark by clicking on the "stop" button.
9. Analyze the capture DCCP packets in wireshark by reviewing the packet details, including the source and destination IP addresses, port numbers, sequence numbers, and payload data.


dccp_trace (1).pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
13 1.753774 192.168.1.31 201.11.59.173 DCCP 342 32772 → 5001 [DataAck] Seq=7 (Ack=4)
14 1.939758 201.11.59.173 192.168.1.31 DCCP 86 5001 → 32772 [Ack] Seq=5 (Ack=6)
15 1.943711 192.168.1.31 201.11.59.173 DCCP 342 32772 → 5001 [DataAck] Seq=8 (Ack=5)
16 2.131548 201.11.59.173 192.168.1.31 DCCP 94 5001 → 32772 [Ack] Seq=6 (Ack=7)
17 2.131654 192.168.1.31 201.11.59.173 DCCP 342 32772 → 5001 [DataAck] Seq=9 (Ack=6)
18 2.319741 192.168.1.31 201.11.59.173 DCCP 306 32772 → 5001 [Data] Seq=10
19 2.321494 201.11.59.173 192.168.1.31 DCCP 94 5001 → 32772 [Ack] Seq=7 (Ack=8)
20 2.415755 192.168.1.31 201.11.59.173 DCCP 342 32772 → 5001 [DataAck] Seq=11 (Ack=7)
21 2.510882 201.11.59.173 192.168.1.31 DCCP 94 5001 → 32772 [Ack] Seq=8 (Ack=9)
22 2.510988 192.168.1.31 201.11.59.173 DCCP 342 32772 → 5001 [DataAck] Seq=12 (Ack=8)
23 2.603740 192.168.1.31 201.11.59.173 DCCP 306 32772 → 5001 [Data] Seq=13
24 2.696705 201.11.59.173 192.168.1.31 DCCP 86 5001 → 32772 [Ack] Seq=9 (Ack=10)
25 2.696808 192.168.1.31 201.11.59.173 DCCP 342 32772 → 5001 [DataAck] Seq=14 (Ack=9)
26 2.791767 192.168.1.31 201.11.59.173 DCCP 306 32772 → 5001 [Data] Seq=15
27 2.887774 192.168.1.31 201.11.59.173 DCCP 306 32772 → 5001 [Data] Seq=16
Destination Address: 201.11.59.173
Datagram Congestion Control Protocol, Src Port: 32772, Dst Port: 5001 [Data] Seq=13
Source Port: 32772
Destination Port: 5001
[Stream index: 0]
Data Offset: 4
CCVal: 12
Checksum Coverage: 0
Checksum: 0x4d3a [correct]
[Checksum Status: Good]
Type: Data (2)
Extended Sequence Numbers: True
Sequence Number: 13 (relative sequence number)
Sequence Number (raw): 17867828713
Data (256 bytes)
Data: 2021222324252627282a2b2c2d2e2f303132333435363738393a3b3c3d3e3f40414243…
[Length: 256]
0000 00 d0 41 0a f2 6b 00 b0 d0 35 15 bf 08 00 45 00 ··A··k·· ·5····E·
0010 01 24 15 13 40 00 40 21 5e 26 c0 a8 01 1f c9 0b ·$··@·@! ^&····
0020 3b ad 80 04 13 89 04 c0 4d 3a 05 00 00 04 29 01 ;····· M:····)·
0030 6d e9 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d m· !"#$% &'()*+,-
0040 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d ./012345 6789:;<=
0050 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d >?@ABCDE FGHIJKLM
0060 4e 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d NOPQRSTU VWXYZ[\]
0070 5e 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d ^_`abcde fghijklm
0080 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d nopqrstu vwxyz{|}
0090 7e 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e ~ !"#$%& '()*+,-.
00a0 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e /0123456 789:;<=>
00b0 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e ?@ABCDEF GHIJKLMN
00c0 4f 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e OPQRSTUV WXYZ[\]^
00d0 5f 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e _`abcdef ghijklmn
00e0 6f 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e opqrstuv wxyz{|}~
00f0 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f !"#$%&' ()*+,-./
0100 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 01234567 89:;<=>?
0110 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f @ABCDEFG HIJKLMNO
0120 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f PQRSTUVW XYZ[\]^_
0130 60 61 `a
dccp_trace (1).pcap
Packets: 5060 · Displayed: 5060 (100.0%)
Profile: Default
ENG IN
1:54 AM 4/11/2023

__/__/____

DCCP is basically a message-based transport-level protocol. The setting of a secure connection is easily maintained using it, its closure i.e ECN (Explicit Congestion Notification), cogestion control, and negotiation of features. DCCP is a great technique to access congestion control mechanisms, also we don't need to implement them at the application level also. DCCP basically allows similar Transfer Control Protocol feeds also, but delivery in the order of transmission cannot be done.

DCCP connection setup can be explained through the below image,



Features of DCCP:

1. DCCP is a non-reliable datagram stream, with a good feature of confirmation.
2. DCCP helps to secure negotiation of options, including negotiation of the most suitable mechanism for congestion control.

__/__/____

3. It provides a secure handshake protocol with the purpose of initializing and closing the connection of DCCP.
4. It plays a vital role in the discovery of the maximum transmitting unit on the chosen path by the user.
5. It provides technique that allows sever to avoid storing states for attempted unconnected, unconfirmed disconnections, or for already closed connections as well.
6. Confirmation mechanisms are a very good feature of DCCP which helps to communicate packet loss and ECN information.

Advantages :

- Congestion Control - DCCP includes congestion control algorithms that helps to prevent n/w overload & ensure reliable delivery of data.
- Quality of Service (QOS) support: It is very useful for applications such as video streaming or voice over IP, where low latency and high reliability are essential.
- Flexibility: DCCP is designed to be a flexible protocol, allowing network administrators to choose from a variety of congestion control algorithms based on the specific requirements of their network and applications.
- Compatibility: DCCP is designed to work with existing IP networks and is compatible with traditional IP protocols like TCP and UDP.

__/__/____

Analysis of flow (I/O graph)

- In the flow graph, after applying a filter to display only DCCP traffic, we can see the I/O graph of only the required protocol.

- In the given graph x represents time in seconds and y represents packets/seconds.

The constant flow of packets shows a stable connection. Number of packets/sec increases for 0 to 5 sec. At 5 second the flatlines denoting a stable connection. When connection at t = 23 seconds, number of packet/second start to fall to 0, ending at 25th sec, denoting end of communication between source & destination.

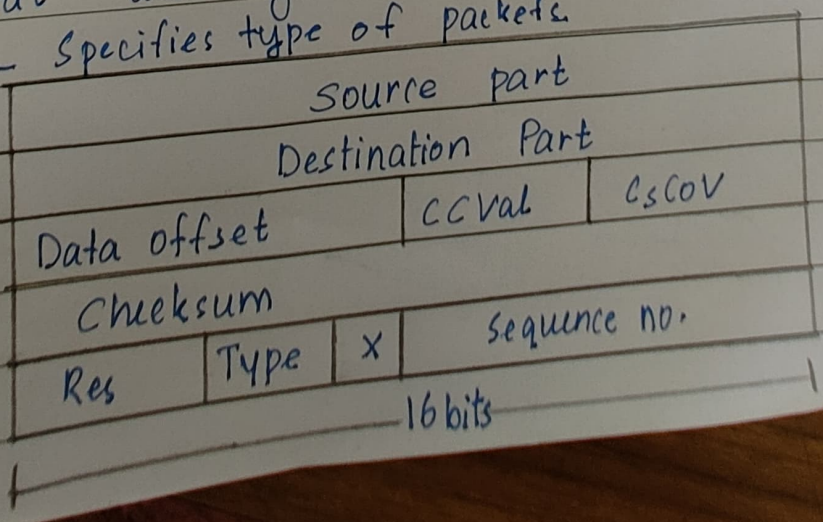
Protocol Hierarchy of DCCP:

First, looking at DCCP packets general header, image attached, it consists of source port, destination port, data offset, CC val, checksum coverage, checksum, type, sequencenumber (extended).

Data offset - offset from start of packet DCCP header to start it's application data area.

CC val - used by HC-sender CCID.

Type - Specifies type of packets.

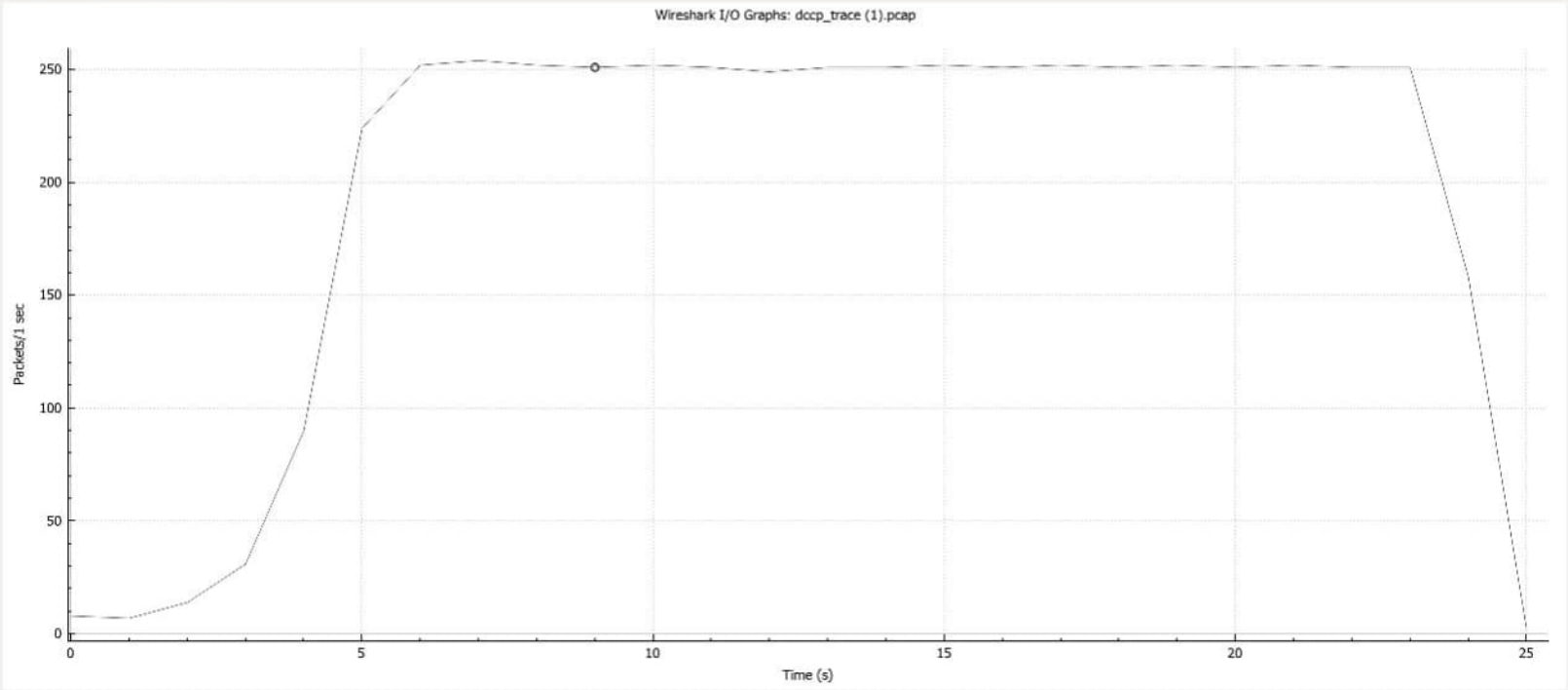


__/__/____

X value determines length of sequence & acknowledgement numbers
if X = 1 ⇒ extended generic header of 48 bit sequence.

Sequence Number - identifier packet uniquely in sequence of all packets sent by source.

Wireshark · I/O Graphs · dccp_trace (1).pcap



Click to select packet 1383 (y = 251).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period	Y Axis Factor
☐	TCP Errors	tcp.analysis.flags		Bar	Packets		None	1
☑	DCCP			Line	Packets		None	1

+ − Mouse ◉ drags ○ zooms Interval 1 sec ☐ Time of day ☐ Log scale ☑ Automatic Update Reset

Save As... Copy Copy from Close Help

ENG IN 1:53 AM 4/11/2023

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	5060	100.0	1537118	481 k	0	0	0	5060
Ethernet	100.0	5060	4.6	70858	22 k	0	0	0	5060
Internet Protocol Version 4	100.0	5058	6.6	101160	31 k	0	0	0	5058
Datagram Congestion Control Protocol	100.0	5058	88.8	1365044	427 k	58	3148	986	5058
Data	98.8	5000	83.3	1280000	401 k	5000	1280000	401 k	5000
Address Resolution Protocol	0.0	2	0.0	74	23	2	74	23	2

__/__/____

e. The structure of wireshark consists of several components.

1. Capture Interfaces: Wireshark supports various network capture interfaces, including ethernet, wi-fi and Bluetooth. These interfaces allow wireshark to capture packets as they traverse the network.
2. Packet List Pane: The packet list pane consists, displays a list of all captured packets. Each packet is displayed in a separate row & include details such as the source and destination address, protocol, length & time of capture.
3. Packet Detail Pane: The packet detail pane displays detailed information about the selected packet in the packet line pane. This pane includes information such as the header and payload of the packet, along with information about the specific protocol used.
4. Packet Bytes Pane: The packet bytes pane displays the raw bytes of the selected packets. This pane is useful for analyzing the raw data of a packet & for debugging network issues.
5. Display Filters: Wireshark allows user to apply display filters to narrow down the packets displayed in the packet list pane. Filters can be applied based on criteria such as protocol, source address, destination address or packet length.
6. Statistics: Wireshark provides a range of statistics & graphs that can be used to analyze network traffic patterns.
7. Capture Filters: Wireshark also allows users to apply capture filters to specify the specific types of packets that should be captured. This can be useful for capturing only specific types of traffic such as HTTP or DNS traffic.