Tirth V Patel (200435378)

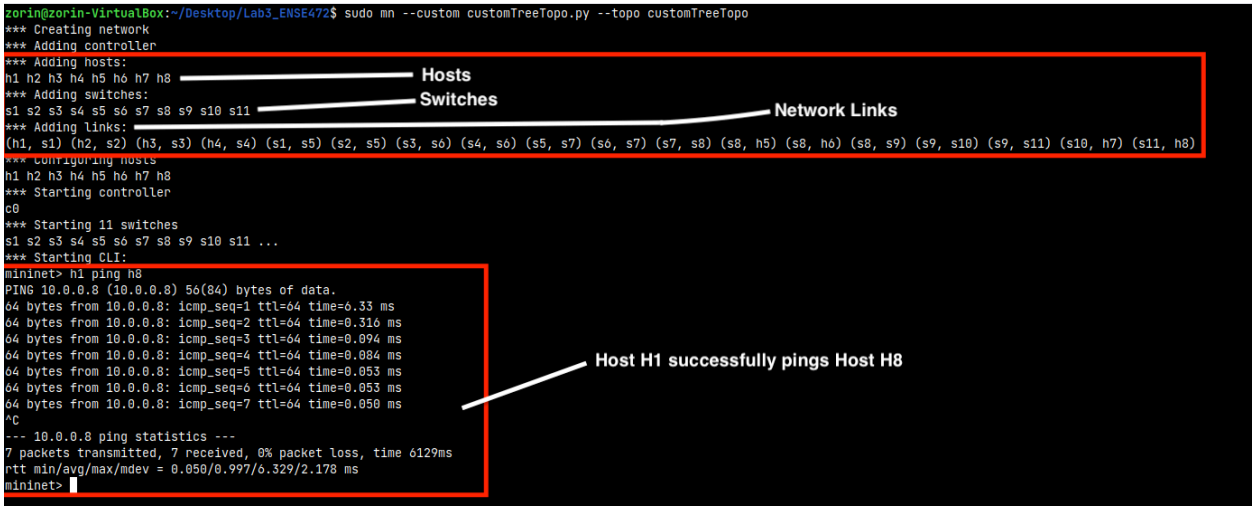
Ramanpreet Singh (200384219)

ENSE 472

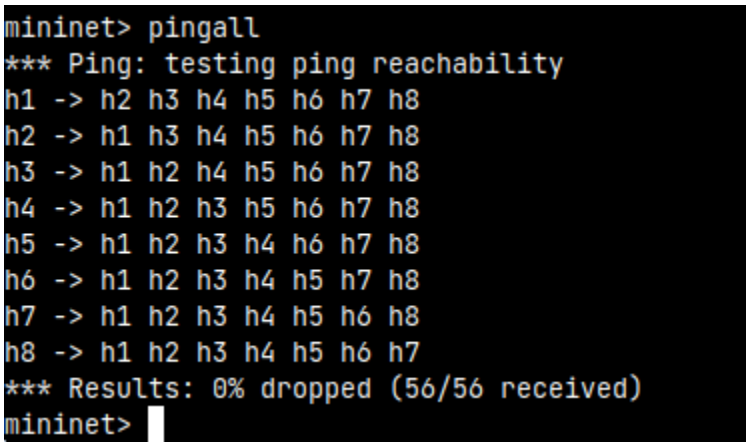
Lab: 3

Q 1]. Create a mininet simulation and verify connectivity :

The custom Mininet topology was created using customTreeTopo.py, which configured 8 hosts (h1 to h8) and 11 switches (s1 to s11) with the default Mininet controller. The following screenshot shows that the h1 to h8 ping operation was successful, with 7 packets transmitted and received and 0 packet loss.



Verify Connectivity: As shown in the following screenshot, the pingall command executed successfully, and all hosts were reachable, confirming successful connectivity.

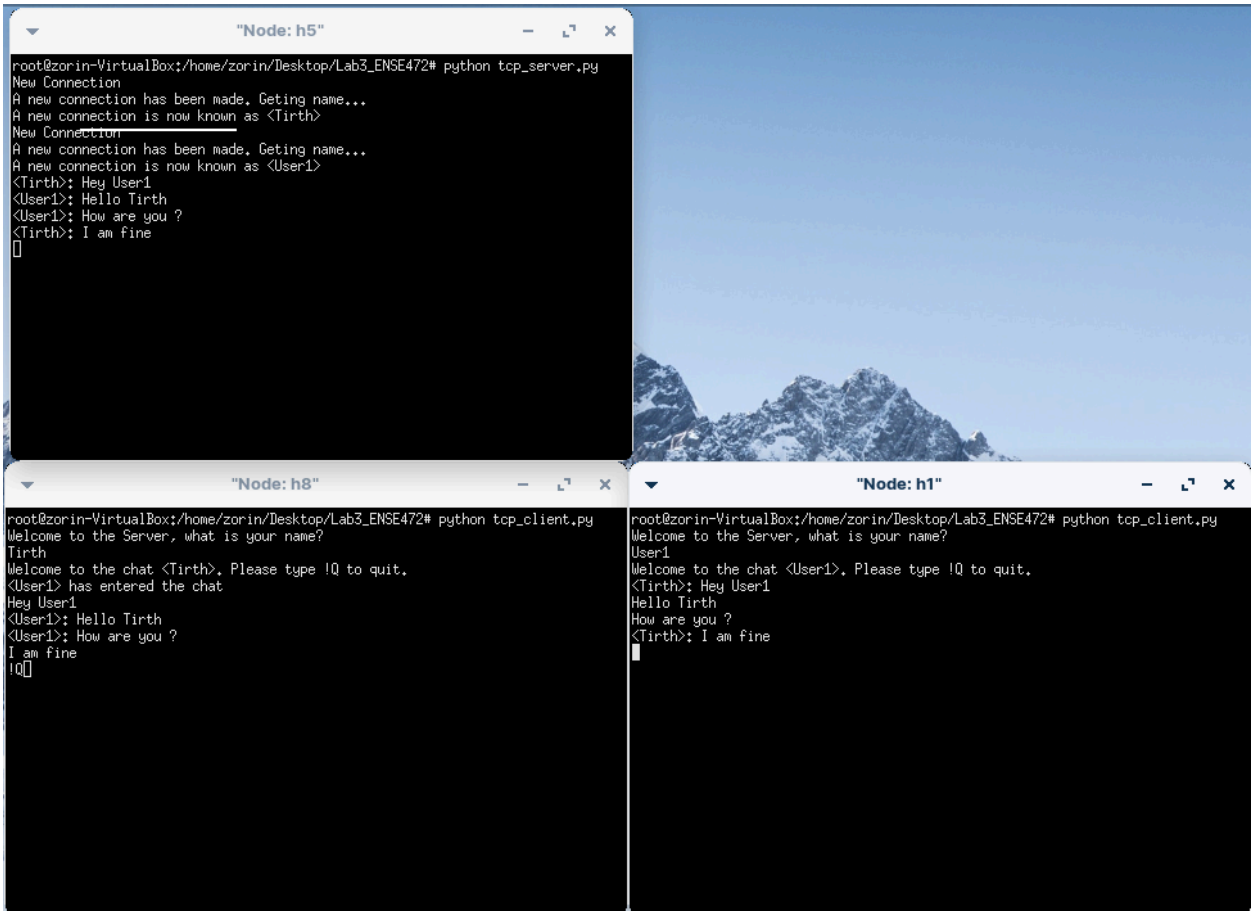


Dump: In the screenshot below, the dump command in Mininet provides detailed information about all network entities in the topology. It displays the class, IP address, and Process ID (PID) for each host, switch, and controller. Each host is shown with its IP (10.0.0.x) and interface (eth0), while switches and the controller are displayed with loopback addresses (127.0.0.1) and their respective PIDs.

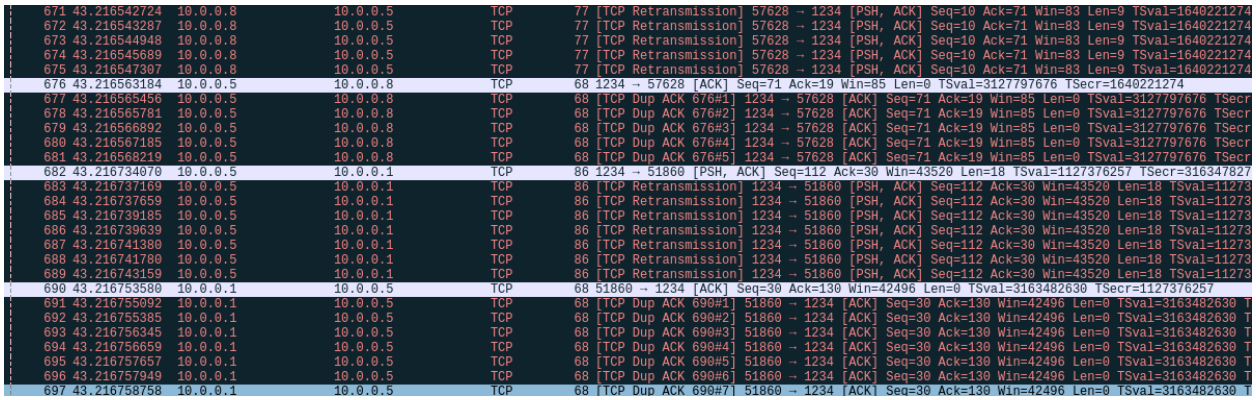
```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=2529>
<Host h2: h2-eth0:10.0.0.2 pid=2531>
<Host h3: h3-eth0:10.0.0.3 pid=2533>
<Host h4: h4-eth0:10.0.0.4 pid=2535>
<Host h5: h5-eth0:10.0.0.5 pid=2537>
<Host h6: h6-eth0:10.0.0.6 pid=2539>
<Host h7: h7-eth0:10.0.0.7 pid=2541>
<Host h8: h8-eth0:10.0.0.8 pid=2543>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None pid=2548>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth2:None pid=2551>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth2:None pid=2554>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None pid=2557>
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth2:None,s5-eth3:None pid=2560>
<OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth2:None,s6-eth3:None pid=2563>
<OVSSwitch s7: lo:127.0.0.1,s7-eth1:None,s7-eth2:None,s7-eth3:None pid=2566>
<OVSSwitch s8: lo:127.0.0.1,s8-eth1:None,s8-eth2:None,s8-eth3:None,s8-eth4:None pid=2569>
<OVSSwitch s9: lo:127.0.0.1,s9-eth1:None,s9-eth2:None,s9-eth3:None pid=2572>
<OVSSwitch s10: lo:127.0.0.1,s10-eth1:None,s10-eth2:None pid=2575>
<OVSSwitch s11: lo:127.0.0.1,s11-eth1:None,s11-eth2:None pid=2578>
<Controller c0: 127.0.0.1:6653 pid=2522>
mininet>
```

Provide the class, IP information, and PID for the hosts, switches, and controller

Q 2]. Load the TCP-based chat server from lab 2 with the server on h5, and clients on h1 and h8. Using the mininet controller for routing, and wireshark for testing, see if either h4 and/or h6 can see the chat packets being sent and read their contents.



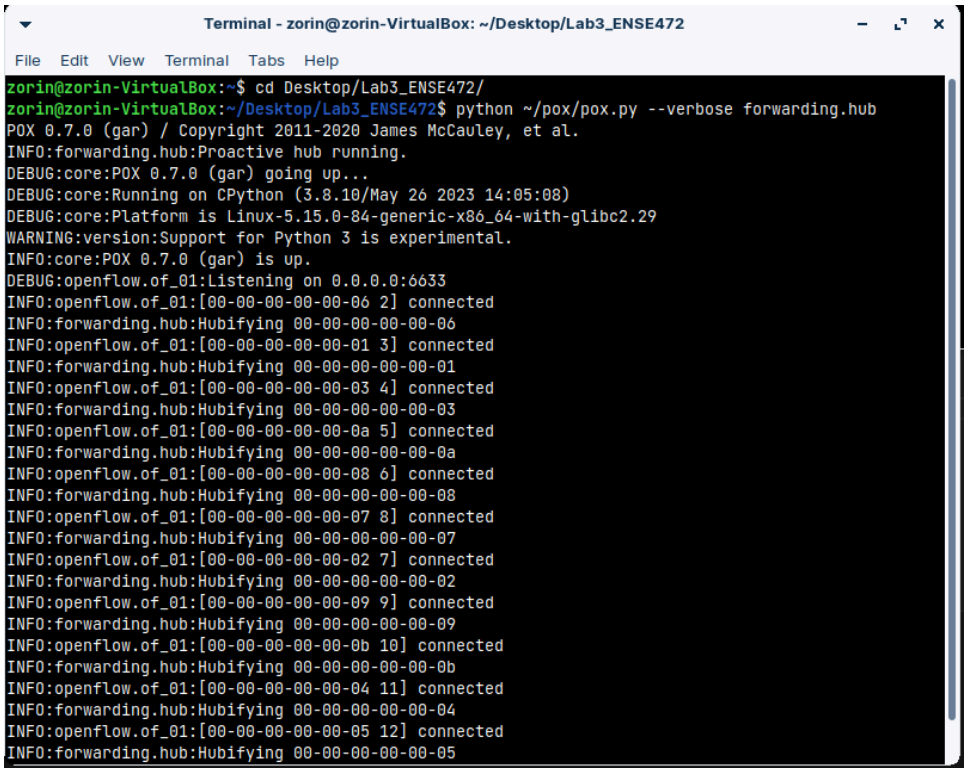
The above image shows a TCP-based chat server running on host **h5**, with clients on hosts **h1** and **h8** successfully connecting and exchanging messages.



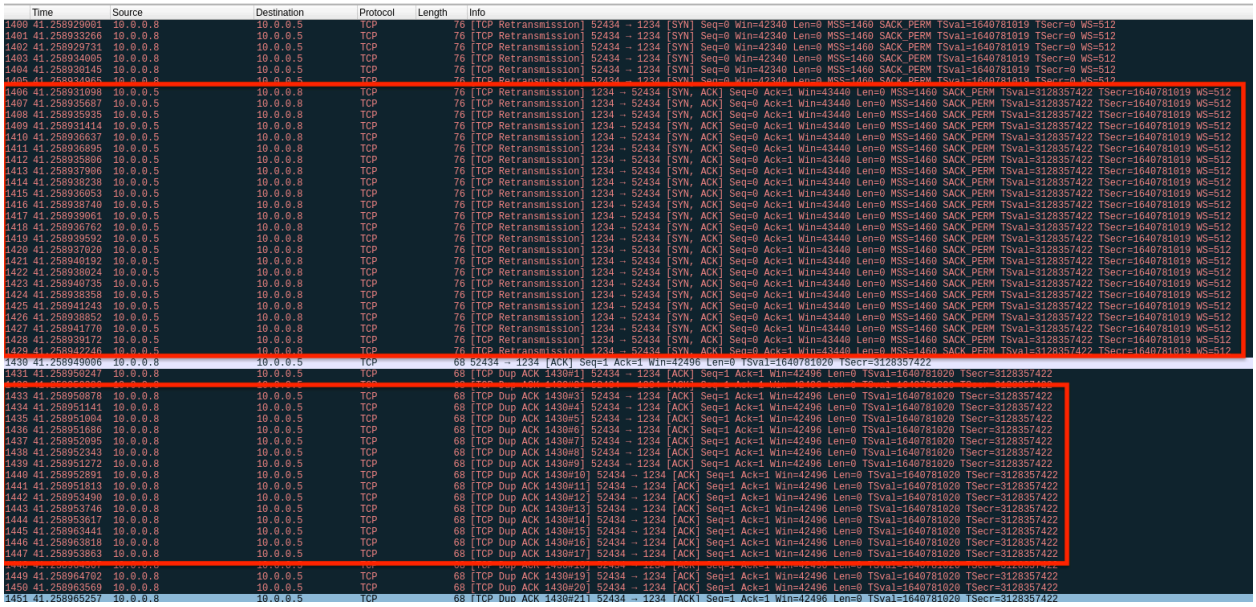
671	43.216542724	10.0.0.8	10.0.0.5	TCP	77 [TCP Retransmission] 57628 → 1234 [PSH, ACK] Seq=10 Ack=71 Win=83 Len=9 TSval=1640221274
672	43.216543287	10.0.0.8	10.0.0.5	TCP	77 [TCP Retransmission] 57628 → 1234 [PSH, ACK] Seq=10 Ack=71 Win=83 Len=9 TSval=1640221274
673	43.216544948	10.0.0.8	10.0.0.5	TCP	77 [TCP Retransmission] 57628 → 1234 [PSH, ACK] Seq=10 Ack=71 Win=83 Len=9 TSval=1640221274
674	43.216545689	10.0.0.8	10.0.0.5	TCP	77 [TCP Retransmission] 57628 → 1234 [PSH, ACK] Seq=10 Ack=71 Win=83 Len=9 TSval=1640221274
675	43.216547307	10.0.0.8	10.0.0.5	TCP	77 [TCP Retransmission] 57628 → 1234 [PSH, ACK] Seq=10 Ack=71 Win=83 Len=9 TSval=1640221274
676	43.216563184	10.0.0.5	10.0.0.8	TCP	68 1234 → 57628 [ACK] Seq=71 Ack=19 Win=85 Len=0 TSval=3127797676 TSecr=1640221274
677	43.216565456	10.0.0.5	10.0.0.8	TCP	68 [TCP Dup ACK 676#1] 1234 → 57628 [ACK] Seq=71 Ack=19 Win=85 Len=0 TSval=3127797676 TSecr
678	43.216565781	10.0.0.5	10.0.0.8	TCP	68 [TCP Dup ACK 676#2] 1234 → 57628 [ACK] Seq=71 Ack=19 Win=85 Len=0 TSval=3127797676 TSecr
679	43.216566892	10.0.0.5	10.0.0.8	TCP	68 [TCP Dup ACK 676#3] 1234 → 57628 [ACK] Seq=71 Ack=19 Win=85 Len=0 TSval=3127797676 TSecr
680	43.216567185	10.0.0.5	10.0.0.8	TCP	68 [TCP Dup ACK 676#4] 1234 → 57628 [ACK] Seq=71 Ack=19 Win=85 Len=0 TSval=3127797676 TSecr
681	43.216568219	10.0.0.5	10.0.0.8	TCP	68 [TCP Dup ACK 676#5] 1234 → 57628 [ACK] Seq=71 Ack=19 Win=85 Len=0 TSval=3127797676 TSecr
682	43.216734070	10.0.0.5	10.0.0.1	TCP	86 1234 → 51860 [PSH, ACK] Seq=112 Ack=30 Win=43520 Len=18 TSval=1127376257 TSecr=316347827
683	43.216737169	10.0.0.5	10.0.0.1	TCP	86 [TCP Retransmission] 1234 → 51860 [PSH, ACK] Seq=112 Ack=30 Win=43520 Len=18 TSval=11273
684	43.216737659	10.0.0.5	10.0.0.1	TCP	86 [TCP Retransmission] 1234 → 51860 [PSH, ACK] Seq=112 Ack=30 Win=43520 Len=18 TSval=11273
685	43.216739185	10.0.0.5	10.0.0.1	TCP	86 [TCP Retransmission] 1234 → 51860 [PSH, ACK] Seq=112 Ack=30 Win=43520 Len=18 TSval=11273
686	43.216739639	10.0.0.5	10.0.0.1	TCP	86 [TCP Retransmission] 1234 → 51860 [PSH, ACK] Seq=112 Ack=30 Win=43520 Len=18 TSval=11273
687	43.216741380	10.0.0.5	10.0.0.1	TCP	86 [TCP Retransmission] 1234 → 51860 [PSH, ACK] Seq=112 Ack=30 Win=43520 Len=18 TSval=11273
688	43.216741780	10.0.0.5	10.0.0.1	TCP	86 [TCP Retransmission] 1234 → 51860 [PSH, ACK] Seq=112 Ack=30 Win=43520 Len=18 TSval=11273
689	43.216743159	10.0.0.5	10.0.0.1	TCP	86 [TCP Retransmission] 1234 → 51860 [PSH, ACK] Seq=112 Ack=30 Win=43520 Len=18 TSval=11273
690	43.216753580	10.0.0.1	10.0.0.5	TCP	68 51860 → 1234 [ACK] Seq=30 Ack=130 Win=42496 Len=0 TSval=3163482630 TSecr=1127376257
691	43.216755092	10.0.0.1	10.0.0.5	TCP	68 [TCP Dup ACK 690#1] 51860 → 1234 [ACK] Seq=30 Ack=130 Win=42496 Len=0 TSval=3163482630 T
692	43.216755385	10.0.0.1	10.0.0.5	TCP	68 [TCP Dup ACK 690#2] 51860 → 1234 [ACK] Seq=30 Ack=130 Win=42496 Len=0 TSval=3163482630 T
693	43.216756345	10.0.0.1	10.0.0.5	TCP	68 [TCP Dup ACK 690#3] 51860 → 1234 [ACK] Seq=30 Ack=130 Win=42496 Len=0 TSval=3163482630 T
694	43.216756659	10.0.0.1	10.0.0.5	TCP	68 [TCP Dup ACK 690#4] 51860 → 1234 [ACK] Seq=30 Ack=130 Win=42496 Len=0 TSval=3163482630 T
695	43.216757657	10.0.0.1	10.0.0.5	TCP	68 [TCP Dup ACK 690#5] 51860 → 1234 [ACK] Seq=30 Ack=130 Win=42496 Len=0 TSval=3163482630 T
696	43.216757949	10.0.0.1	10.0.0.5	TCP	68 [TCP Dup ACK 690#6] 51860 → 1234 [ACK] Seq=30 Ack=130 Win=42496 Len=0 TSval=3163482630 T
697	43.216758758	10.0.0.1	10.0.0.5	TCP	68 [TCP Dup ACK 690#7] 51860 → 1234 [ACK] Seq=30 Ack=130 Win=42496 Len=0 TSval=3163482630 T

The above image shows captured packets in Wireshark, indicating that the TCP-based chat packets are only exchanged between h1, h5, and h8. There is no evidence of packets being delivered to h4 or h6, as the switches are acting as smart switches, only delivering packets to their intended destinations. This confirms that h4 and h6 cannot see or read the chat messages in this setup.

Q 3]. Replace the mininet controller with the POX Flooding algorithm. Do the TCP-based chat packets still arrive at their destination? Are there any broadcast storms?



The above image shows the POX controller running with the flooding algorithm (forwarding.hub), where each switch is connected and set to "hubify," meaning it forwards incoming packets to all ports.

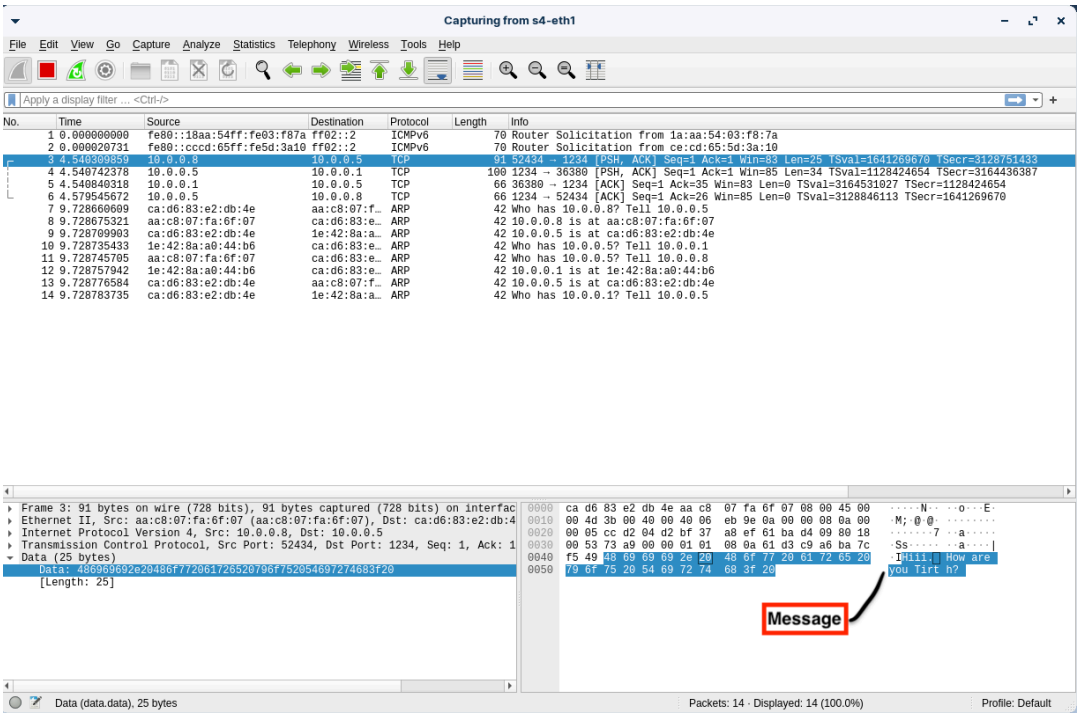


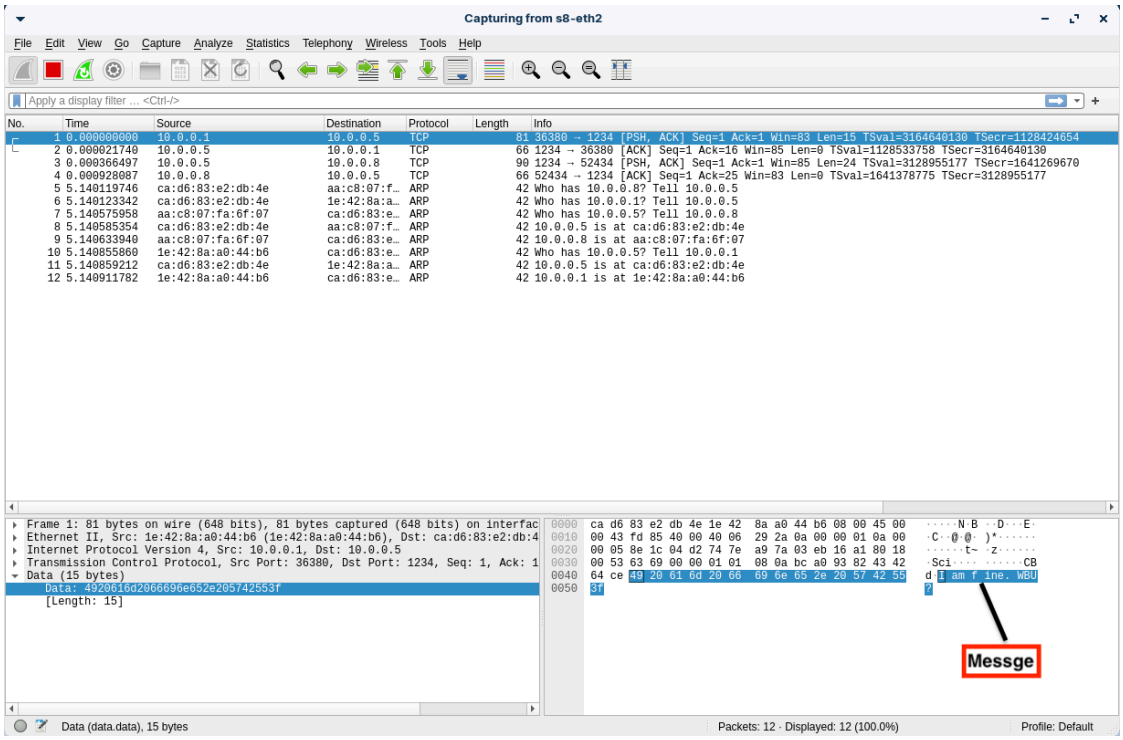
	Time	Source	Destination	Protocol	Length	Info
1400	41.258929001	10.0.0.8	10.0.0.5	TCP	76	[TCP Retransmission] 52434 → 1234 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=1640781019 TSecr=0 WS=512
1401	41.258933266	10.0.0.8	10.0.0.5	TCP	76	[TCP Retransmission] 52434 → 1234 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=1640781019 TSecr=0 WS=512
1402	41.258929731	10.0.0.8	10.0.0.5	TCP	76	[TCP Retransmission] 52434 → 1234 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=1640781019 TSecr=0 WS=512
1403	41.258934005	10.0.0.8	10.0.0.5	TCP	76	[TCP Retransmission] 52434 → 1234 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=1640781019 TSecr=0 WS=512
1404	41.258930145	10.0.0.8	10.0.0.5	TCP	76	[TCP Retransmission] 52434 → 1234 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=1640781019 TSecr=0 WS=512
1405	41.258934065	10.0.0.8	10.0.0.5	TCP	76	[TCP Retransmission] 52434 → 1234 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM TSval=1640781019 TSecr=0 WS=512
1406	41.258931098	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1407	41.258935687	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1408	41.258935935	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1409	41.258931414	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1410	41.258936637	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1411	41.258936895	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1412	41.258935806	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1413	41.258937906	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1414	41.258938238	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1415	41.258936053	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1416	41.258938740	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1417	41.258939061	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1418	41.258936762	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1419	41.258939592	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1420	41.258937020	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1421	41.258940192	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1422	41.258938024	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1423	41.258940735	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1424	41.258938358	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1425	41.258941243	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1426	41.258938852	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1427	41.258941170	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1428	41.258939172	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1429	41.258942246	10.0.0.5	10.0.0.8	TCP	76	[TCP Retransmission] 1234 → 52434 [SYN, ACK] Seq=0 Ack=1 Win=43440 Len=0 MSS=1460 SACK_PERM TSval=3128357422 TSecr=1640781019 WS=512
1430	41.258949006	10.0.0.8	10.0.0.5	TCP	68	52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1431	41.258950247	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#1] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1432	41.258952864	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#2] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1433	41.258950878	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#3] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1434	41.258951141	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#4] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1435	41.258951004	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#5] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1436	41.258951686	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#6] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1437	41.258952095	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#7] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1438	41.258952343	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#8] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1439	41.258951272	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#9] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1440	41.258952891	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#10] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1441	41.258951813	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#11] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1442	41.258953490	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#12] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1443	41.258953746	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#13] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1444	41.258953617	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#14] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1445	41.258963441	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#15] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1446	41.258963818	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#16] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1447	41.258953863	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#17] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1448	41.258964702	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#18] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1449	41.258964702	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#19] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1450	41.258963569	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#20] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422
1451	41.258965257	10.0.0.8	10.0.0.5	TCP	68	[TCP Dup ACK 1430#21] 52434 → 1234 [ACK] Seq=1 Ack=1 Win=42496 Len=0 TSval=1640781020 TSecr=3128357422

The TCP-based chat packets are still reaching their destination (10.0.0.5 and 10.0.0.8). However, the presence of multiple retransmissions and duplicate acknowledgments (Dup ACK) suggests network congestion or broadcast storms.

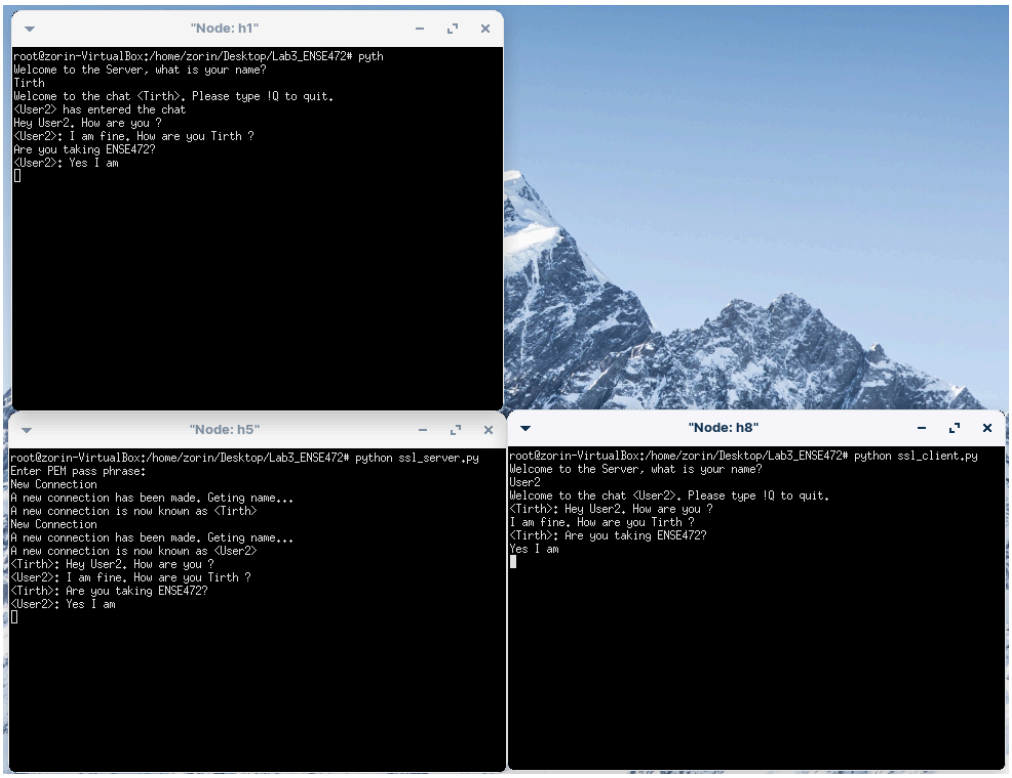
Q 4]. With the flooding controller, verify if either h4 and/or h6 can see the chat packets being sent and read their contents.

The images below show packet captures on the h4 and h6 interfaces, verifying that, with the flooding controller in use, both h4 and h6 can indeed see the chat packets being sent between h1, h5, and h8. Additionally, they are able to read the contents of the messages.

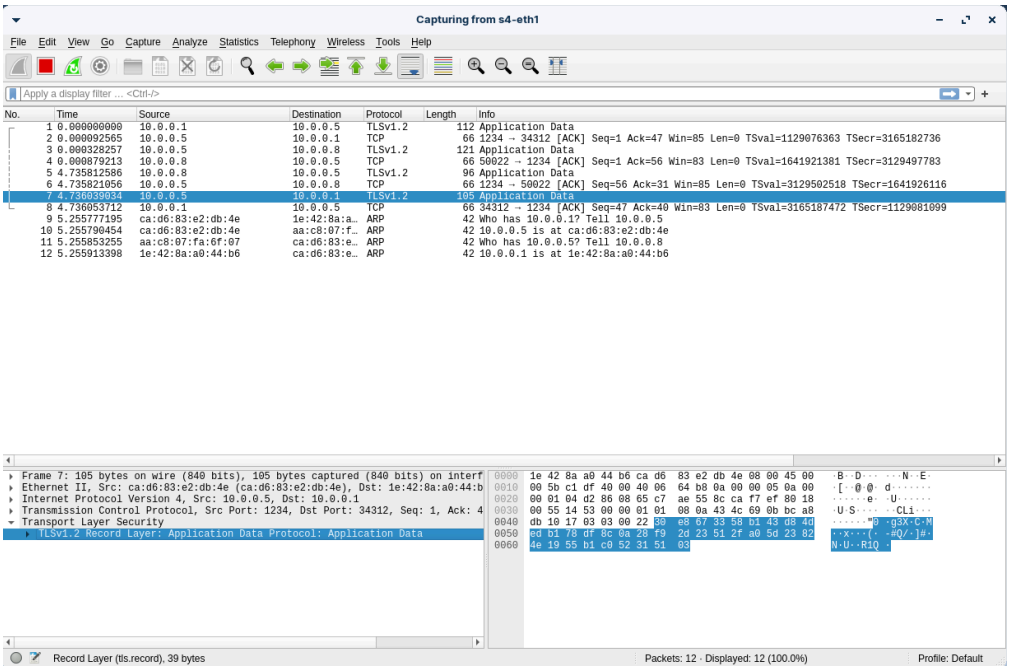

Capturing from s4-eth1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000000 fe80::18aa:54ff:fe03:f87a ff02::2 ICMPv6 70 Router Solicitation from 1a:aa:54:03:f8:7a
2 0.000020731 fe80::cccd:65ff:fe5d:3a10 ff02::2 ICMPv6 70 Router Solicitation from ce:cd:65:5d:3a:10
3 4.540309859 10.0.0.8 10.0.0.5 TCP 91 52434 → 1234 [PSH, ACK] Seq=1 Ack=1 Win=83 Len=25 TSval=1641269670 TSecr=3128751433
4 4.540742378 10.0.0.5 10.0.0.1 TCP 100 1234 → 36380 [PSH, ACK] Seq=1 Ack=1 Win=85 Len=34 TSval=1128424654 TSecr=3164436387
5 4.540840318 10.0.0.1 10.0.0.5 TCP 66 36380 → 1234 [ACK] Seq=1 Ack=35 Win=83 Len=0 TSval=3164531027 TSecr=1128424654
6 4.579545672 10.0.0.5 10.0.0.8 TCP 66 1234 → 52434 [ACK] Seq=1 Ack=26 Win=85 Len=0 TSval=3128846113 TSecr=1641269670
7 9.728660609 ca:d6:83:e2:db:4e aa:c8:07:f… ARP 42 Who has 10.0.0.8? Tell 10.0.0.5
8 9.728675321 aa:c8:07:fa:6f:07 ca:d6:83:e… ARP 42 10.0.0.8 is at aa:c8:07:fa:6f:07
9 9.728709903 ca:d6:83:e2:db:4e 1e:42:8a:a… ARP 42 10.0.0.5 is at ca:d6:83:e2:db:4e
10 9.728735433 1e:42:8a:a0:44:b6 ca:d6:83:e… ARP 42 Who has 10.0.0.5? Tell 10.0.0.1
11 9.728745705 aa:c8:07:fa:6f:07 ca:d6:83:e… ARP 42 Who has 10.0.0.5? Tell 10.0.0.8
12 9.728757942 1e:42:8a:a0:44:b6 ca:d6:83:e… ARP 42 10.0.0.1 is at 1e:42:8a:a0:44:b6
13 9.728776584 ca:d6:83:e2:db:4e aa:c8:07:f… ARP 42 10.0.0.5 is at ca:d6:83:e2:db:4e
14 9.728783735 ca:d6:83:e2:db:4e 1e:42:8a:a… ARP 42 Who has 10.0.0.1? Tell 10.0.0.5
Frame 3: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interfac
Ethernet II, Src: aa:c8:07:fa:6f:07 (aa:c8:07:fa:6f:07), Dst: ca:d6:83:e2:db:4
Internet Protocol Version 4, Src: 10.0.0.8, Dst: 10.0.0.5
Transmission Control Protocol, Src Port: 52434, Dst Port: 1234, Seq: 1, Ack: 1
Data (25 bytes)
Data: 486969692e20486f7720617265207 96f7520546972746 83f20
[Length: 25]
0000 ca d6 83 e2 db 4e aa c8 07 fa 6f 07 08 00 45 00
0010 00 4d 3b 00 40 00 40 06 eb 9e 0a 00 00 08 0a 00
0020 00 05 cc d2 04 d2 bf 37 a8 ef 61 ba d4 09 80 18
0030 00 53 73 a9 00 00 01 01 08 0a 61 d3 c9 a6 ba 7c
0040 f5 49 48 69 69 2e 20 48 6f 77 20 61 72 65 20
0050 79 6f 75 20 54 69 72 74 68 3f 20
Hiii. How are
you Tirt h?
Message
Data (data.data), 25 bytes
Packets: 14 · Displayed: 14 (100.0%)
Profile: Default

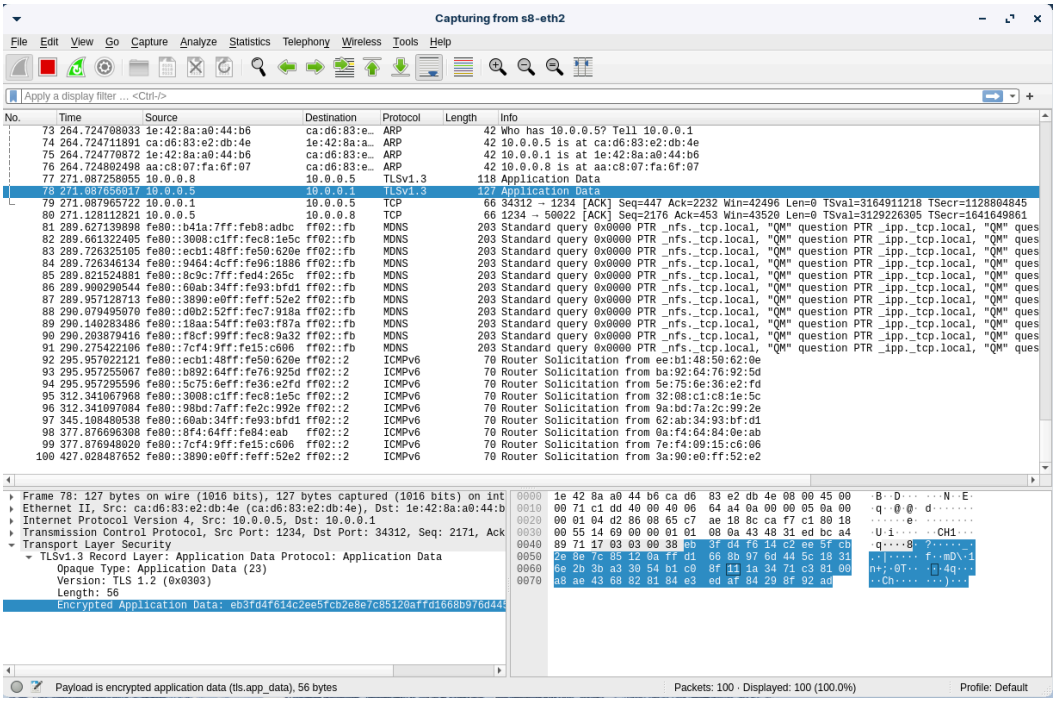

Capturing from s8-eth2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000000 10.0.0.1 10.0.0.5 TCP 81 36380 → 1234 [PSH, ACK] Seq=1 Ack=1 Win=83 Len=15 TSval=3164640130 TSecr=1128424654
2 0.000021740 10.0.0.5 10.0.0.1 TCP 66 1234 → 36380 [ACK] Seq=1 Ack=16 Win=85 Len=0 TSval=1128533758 TSecr=3164640130
3 0.000366497 10.0.0.5 10.0.0.8 TCP 90 1234 → 52434 [PSH, ACK] Seq=1 Ack=1 Win=85 Len=24 TSval=3128955177 TSecr=1641269670
4 0.000928087 10.0.0.8 10.0.0.5 TCP 66 52434 → 1234 [ACK] Seq=1 Ack=25 Win=83 Len=0 TSval=1641378775 TSecr=3128955177
5 5.140119746 ca:d6:83:e2:db:4e aa:c8:07:f… ARP 42 Who has 10.0.0.8? Tell 10.0.0.5
6 5.140123342 ca:d6:83:e2:db:4e 1e:42:8a:a… ARP 42 Who has 10.0.0.1? Tell 10.0.0.5
7 5.140575958 aa:c8:07:fa:6f:07 ca:d6:83:e… ARP 42 Who has 10.0.0.5? Tell 10.0.0.8
8 5.140585354 ca:d6:83:e2:db:4e aa:c8:07:f… ARP 42 10.0.0.5 is at ca:d6:83:e2:db:4e
9 5.140633940 aa:c8:07:fa:6f:07 ca:d6:83:e… ARP 42 10.0.0.8 is at aa:c8:07:fa:6f:07
10 5.140855860 1e:42:8a:a0:44:b6 ca:d6:83:e… ARP 42 Who has 10.0.0.5? Tell 10.0.0.1
11 5.140859212 ca:d6:83:e2:db:4e 1e:42:8a:a… ARP 42 10.0.0.5 is at ca:d6:83:e2:db:4e
12 5.140911782 1e:42:8a:a0:44:b6 ca:d6:83:e… ARP 42 10.0.0.1 is at 1e:42:8a:a0:44:b6
Frame 1: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interfac
Ethernet II, Src: 1e:42:8a:a0:44:b6 (1e:42:8a:a0:44:b6), Dst: ca:d6:83:e2:db:4
Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.5
Transmission Control Protocol, Src Port: 36380, Dst Port: 1234, Seq: 1, Ack: 1
Data (15 bytes)
Data: 4920616d2066696e652e205742553f
[Length: 15]
0000 ca d6 83 e2 db 4e 1e 42 8a a0 44 b6 08 00 45 00
0010 00 43 fd 85 40 00 40 06 29 2a 0a 00 00 01 0a 00
0020 00 05 8e 1c 04 d2 74 7e a9 7a 03 eb 16 a1 80 18
0030 00 53 63 69 00 00 01 01 08 0a bc a0 93 82 43 42
0040 64 ce 49 20 61 6d 20 66 69 6e 65 2e 20 57 42 55
0050 3f
I am f ine. WBU
?
Messge
Data (data.data), 15 bytes
Packets: 12 · Displayed: 12 (100.0%)
Profile: Default

Q 5]. Finally, using the SSL-based chat protocol from Part 2 of lab 2, using the flooding algorithm. See if either h4 and/or h6 can see the chat packets being sent and read their contents.



The images above show an SSL-based chat protocol setup where the server runs on h5, and clients are connected on h1 and h8.





The images above show packet captures on h4 and h6 interfaces with the flooding algorithm in use. Although packets are broadcast across the network, the SSL-based chat protocol encrypts the data, so even though h4 and h6 receive the packets, they cannot view or read the message contents. The encryption provided by SSL ensures that only the intended participants (h1, h5, and h8) can decrypt and read the messages.