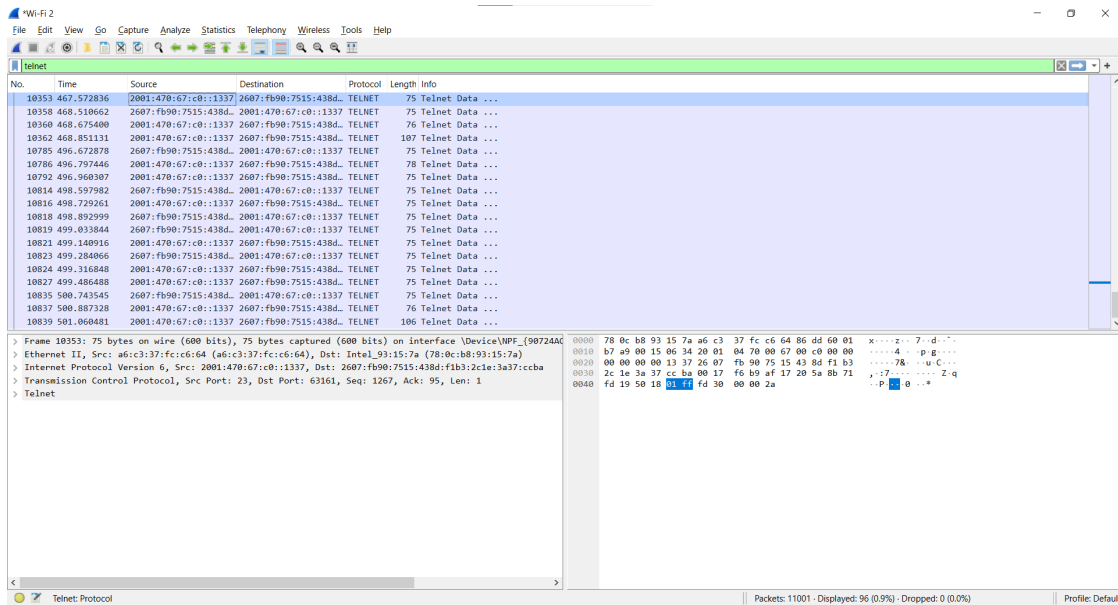


Project:01

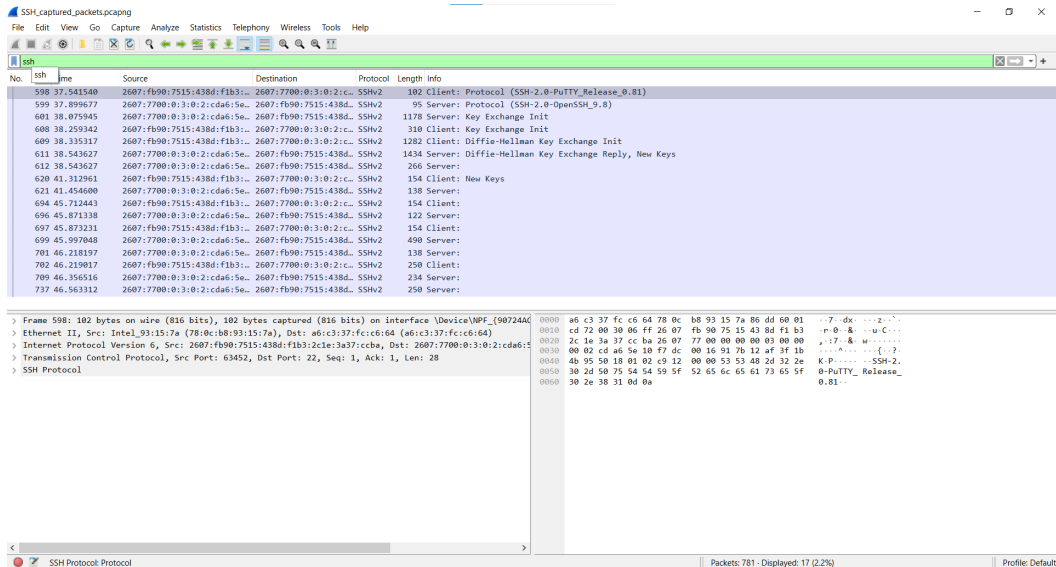
A. What are the IP addresses of **telehack.com** and **sdf.org**?

IP of **telehack.com** - 64.13.139.230



The screenshot shows a Wi-Fi 2 network traffic capture window. The top pane displays a list of captured packets, with the selected packet (No. 10353) showing a Telnet connection from 10.0.0.1 to 64.13.139.230. The bottom pane shows the packet details for the selected Telnet packet, including the Ethernet II header, Internet Protocol Version 6 header, and Transmission Control Protocol header. The packet is identified as a Telnet connection to telehack.com.

IP of **sdf.org** - 205.166.94.16



The screenshot shows an SSH captured_packets.pcapng network traffic capture window. The top pane displays a list of captured packets, with the selected packet (No. 598) showing an SSH connection from 10.0.0.1 to 205.166.94.16. The bottom pane shows the packet details for the selected SSH packet, including the Ethernet II header, Internet Protocol Version 6 header, and SSH Protocol header. The packet is identified as an SSH connection to sdf.org.

B. Screenshots of the packet dump for the TELNET operation and the SSH operation. Please choose the packets with relatively large size (i.e., greater than 300 bytes) so that we can see the data contents.

TELNET packet Dumps

The screenshot displays a Wi-Fi 2 packet capture interface. The packet list on the left shows a TELNET packet (No. 9139) selected. The packet details pane on the right shows the raw data and its ASCII representation. The packet is a TELNET Data packet, 1217 bytes in size, captured on interface \Device\NPF... The packet details pane shows the raw data and its ASCII representation. The packet is a TELNET Data packet, 1217 bytes in size, captured on interface \Device\NPF... The packet details pane shows the raw data and its ASCII representation.

Ssh Packet dumps

The screenshot displays a Wi-Fi 2 packet capture interface. The packet list on the left shows an SSH packet (No. 611) selected. The packet details pane on the right shows the raw data and its ASCII representation. The packet is an SSH Protocol packet, 1434 bytes in size, captured on interface \Device\NPF... The packet details pane shows the raw data and its ASCII representation. The packet is an SSH Protocol packet, 1434 bytes in size, captured on interface \Device\NPF...

C. Please answer, which protocol does PuTTY use to establish encryption key with the SSH server (i.e., which key exchange algorithm is used)? It is okay to consult external sources such as textbooks, online videos, or the web to find the answer to this question ?

The protocol which PuTTY use to establish encryption key with the SSH server (i.e., which key exchange algorithm is used) is the Diffie-Hellman Key Exchange .

This algorithm allows two parties to securely share a secret key over an insecure channel without actually transmitting the key itself.

I personally believe that it is not required to consult external sources such as textbooks, online videos, or the web to find the answer to this question because when you perform the practicals, all the details are mentioned as shown in figure below.

The image shows a Wireshark packet capture of an SSH session. The top pane displays a list of packets, with packet 609 selected. The middle pane shows the details of packet 609, which is an SSH Protocol packet. The bottom pane shows the raw packet data in hexadecimal and ASCII.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
598	37.541540	2607:fb90:7515:438d...	2607:7700:0:3:0:2:c...	SSHv2	102	Client: Protocol (SSH-2.0-PuTTY_Release_0.81)
599	37.899677	2607:7700:0:3:0:2:c...	2607:fb90:7515:438d...	SSHv2	95	Server: Protocol (SSH-2.0-OpenSSH_9.8)
601	38.075945	2607:7700:0:3:0:2:c...	2607:fb90:7515:438d...	SSHv2	1178	Server: Key Exchange Init
608	38.259342	2607:fb90:7515:438d...	2607:7700:0:3:0:2:c...	SSHv2	310	Client: Key Exchange Init
609	38.335317	2607:fb90:7515:438d...	2607:7700:0:3:0:2:c...	SSHv2	1282	Client: Diffie-Hellman Key Exchange Init
611	38.543627	2607:7700:0:3:0:2:c...	2607:fb90:7515:438d...	SSHv2	1434	Server: Diffie-Hellman Key Exchange Reply, New Keys
612	38.543627	2607:7700:0:3:0:2:c...	2607:fb90:7515:438d...	SSHv2	266	Server:
620	41.312961	2607:fb90:7515:438d...	2607:7700:0:3:0:2:c...	SSHv2	154	Client: New Keys
621	41.454600	2607:7700:0:3:0:2:c...	2607:fb90:7515:438d...	SSHv2	138	Server:
694	45.712443	2607:fb90:7515:438d...	2607:7700:0:3:0:2:c...	SSHv2	154	Client:
696	45.871338	2607:7700:0:3:0:2:c...	2607:fb90:7515:438d...	SSHv2	122	Server:
697	45.873231	2607:fb90:7515:438d...	2607:7700:0:3:0:2:c...	SSHv2	154	Client:
699	45.997048	2607:7700:0:3:0:2:c...	2607:fb90:7515:438d...	SSHv2	490	Server:
701	46.218197	2607:7700:0:3:0:2:c...	2607:fb90:7515:438d...	SSHv2	138	Server:
702	46.219017	2607:fb90:7515:438d...	2607:7700:0:3:0:2:c...	SSHv2	250	Client:
709	46.356516	2607:7700:0:3:0:2:c...	2607:fb90:7515:438d...	SSHv2	234	Server:
737	46.563312	2607:7700:0:3:0:2:c...	2607:fb90:7515:438d...	SSHv2	250	Server:

Packet 609 Details:

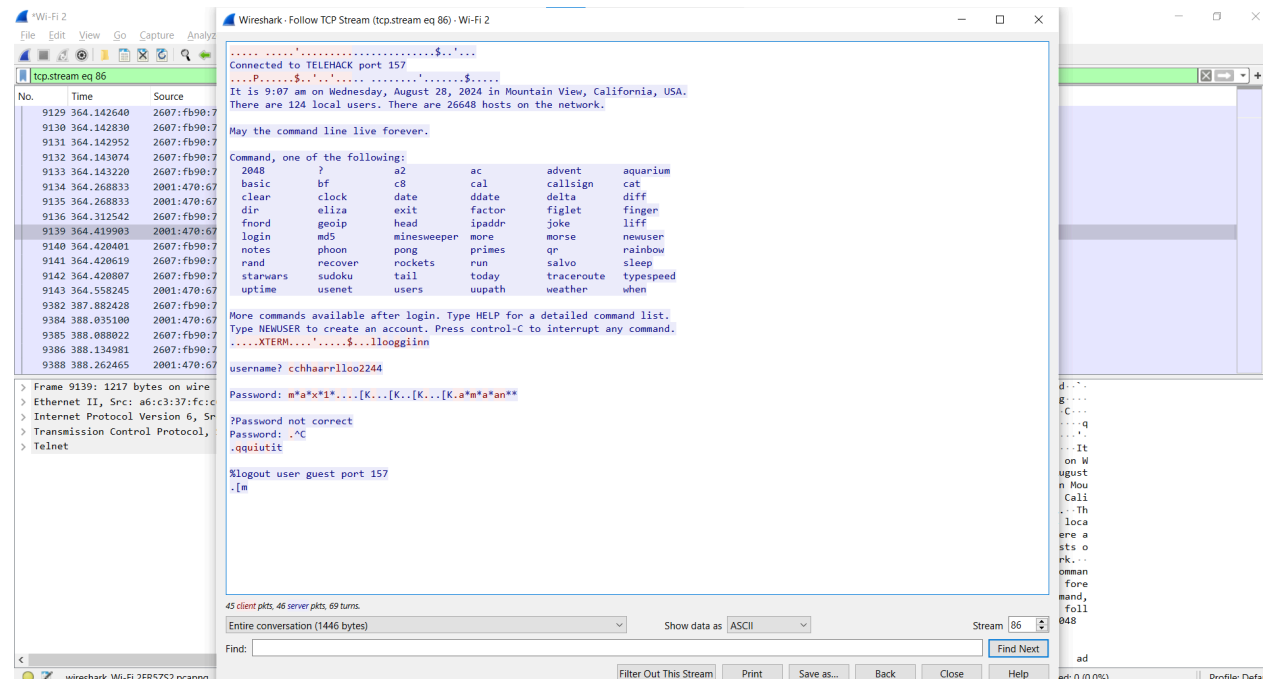
- Frame 609: 1282 bytes on wire (10256 bits), 1282 bytes captured (10256 bits) on interface \Device\NPF_{...}
- Ethernet II, Src: Intel_93:15:7a (78:0c:b8:93:15:7a), Dst: a6:c3:37:fc:c6:64 (a6:c3:37:fc:c6:64)
- Internet Protocol Version 6, Src: 2607:fb90:7515:438d:f1b3:2c1e:3a37:ccb6, Dst: 2607:7700:0:3:0:2:cd6a:5...
- Transmission Control Protocol, Src Port: 63452, Dst Port: 22, Seq: 1645, Ack: 1126, Len: 1208
- SSH Protocol

Raw Packet Data (Hex):

```
0000 a6 c3 37 fc c6 64 78 0c b8 93 15 7a 86 dd 60 01 --7--dx-...z-...-
0010 cd 72 04 cc 06 ff 26 07 fb 90 75 15 43 8d f1 b3 --p-...&-...u-C-...
0020 2c 1e 3a 37 cc ba 26 07 77 00 00 00 00 03 00 00 --;7--&-w-...-...
0030 00 02 cd a6 5e 10 f7 dc 00 16 91 7b 19 1b 3f 1b --...A-...-(-P-...
0040 4f fa 50 18 00 fe dd d6 00 00 00 00 04 b4 08 1e --...P-...-...-...
0050 00 00 04 a6 3f dd 25 d4 e5 9a b0 39 84 0e 82 ff --...?%-...9-...-
0060 7e 24 f3 75 74 29 43 08 af 7c 3c b3 41 11 c7 ca --$-ut)C-...[c-A-...-
0070 06 be 5e a2 5c 08 ad 5b 83 00 77 fd db 9a 9a 1c --^\\-[-...w-...-
0080 e5 37 8c 2a 0f 55 c5 95 a8 26 5c a6 7a 55 7c d9 --7-...U-...&\\zU]-
0090 eb 94 09 ac 71 70 03 fd 93 98 07 09 f5 8f 15 e4 --...qp-...-...-...
00a0 10 00 d8 2c c5 ee 46 3f 3f 32 a9 a3 d7 7c 07 24 --...F? ?2-...]-$
00b0 38 99 41 cd 0b 9a 3d 73 2e 24 a9 b0 ce 72 29 79 8 A-...s-$-...r)y
00c0 5b 5c 6c 28 8a ae d9 a1 f4 1a 9d ad 8b 9d 9b 71 --[\\(-...-...-...q
00d0 d7 1e c4 0c 1e ea 7b 76 82 cc f5 2a 84 0f 15 99 --...{v-...-...-...
00e0 c6 20 8b d2 8b 52 a1 9d bb 3c 3c df 20 c9 37 a0 --...R-...<-...7-...
00f0 8d 7d a8 68 1d 91 a5 4c f9 00 82 c8 4a 97 03 c2 --}h-...L-...-...3-...
0100 d0 ad b9 a3 f7 8a b5 81 4f 73 db 48 ad 8b 09 08 b1 --...-...-...Os-H-...-
0110 f5 8c 85 81 26 68 b8 c6 58 f4 7f 61 77 87 09 71 --...&h-...X-...au-...q
0120 5a 69 a7 1d a3 30 8d 43 49 bd b2 7e ee 36 12 17 Z1-...0 C I-...-6-...
0130 e6 7e 97 ac 23 5c 0b 4a 9e 7d 15 58 56 2c 12 4e --...-...\\-...-}XV-...N
0140 c0 7b 9b 43 75 bd 0b b0 e0 0b 91 d5 8e 18 50 92 --(-C-...-...-...P-...
0150 85 7b a0 b5 5e a3 33 2f f6 07 26 ed fd 9f 23 c9 --(-^A-3/-&-...-...#
0160 79 1f e6 cb e3 ee 5f ca 2f 33 55 1c 61 17 15 c0 y-...-...-/3U-a-...-
```

D. Shortly analyze the packet dump and explain why SSH is more secure than TELNET.

Analyzed packet dump



Why SSH is more secure than Telnet?

SSH: Encrypts all information that's transmitted between the client and the server. This encryption guarantees that indeed on the off chance that the information is captured, it cannot be effortlessly studied or altered by an assailant.

SSH gives information judgment checks utilizing Message Verification Codes (MACs), guaranteeing that the information has not been altered amid transmission. In case the information is changed, the communication session will be ended.

TELNET: Transmits data in plaintext, meaning that all communication, including usernames, passwords, and any commands or data sent over the network, can be easily intercepted and read by anyone who has access to the network traffic.

Telnet does not provide any mechanisms to ensure the integrity of the data being transmitted. This means that data can be altered in transit without detection.

E. Now open the packet capture for the TELNET operations again. You will notice that there are many other types of packets such as DNS, TCP, etc. Please answer: (1) List all different IP addresses that you see in these captured packets; (2) List all the MAC addresses that you see in these captured packets; (3) List all TCP connections between the IP addresses that you capture. Please note that for a TCP connection, you need to provide (Source IP, Source PORT, Destination IP, Destination PORT).

1. All different IP addresses in the captured packets.

The image shows a Wireshark packet capture analysis. The main window displays a list of captured packets. A summary window titled "Wireshark - All Addresses - Wi-Fi 2" is open, showing a table of IP addresses and their associated statistics.

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
All Addresses	2380				0.0046	100%	0.9300	213.397
239.255.255.250	158				0.0003	6.64%	0.0400	47.818
224.0.0.252	1				0.0000	0.04%	0.0100	147.749
224.0.0.251	3				0.0000	0.13%	0.0200	147.745
224.0.0.22	3				0.0000	0.13%	0.0200	147.732
172.253.62.138	1137				0.0022	47.77%	0.9300	213.397
172.20.10.2	2379				0.0046	99.96%	0.9300	213.397
172.20.10.15	1				0.0000	0.04%	0.0100	253.837
172.20.10.1	27				0.0001	1.13%	0.0900	220.727
152.15.38.60	7				0.0000	0.29%	0.0300	200.863
142.250.31.101	1044				0.0020	43.87%	0.1300	473.923

The main packet list shows the following details for the selected packet (No. 9139):

- Time: 364.124952
- Source: 2607:fb90:7515:438d::2
- Destination: 2001:470:67:c0::1337
- Protocol: TELNET
- Length: 77
- Info: Telnet Data ...

The packet details pane shows the following layers:

- Frame 9139: 1217 bytes on wire (9736 bits), 1217 bytes captured (9736 bits) on interface 0
- Ethernet II, Src: a6:c3:37:fc:c6:64 (a6:c3:37:fc:c6:64), Dst: 08:00:27:00:00:00
- Internet Protocol Version 6, Src: 2001:470:67:c0::1337, Dst: 2001:470:67:c0::1337
- Transmission Control Protocol, Src Port: 23, Dst Port: 23
- Telnet

The packet bytes pane shows the raw data in hexadecimal and ASCII format.

2. All different MAC addresses in the captured packets.

The screenshot shows the 'Endpoints - Wi-Fi 2' window in Wireshark. The 'Endpoint Settings' panel on the left has 'Name resolution' and 'Limit to display filter' unchecked. The 'Copy' button is selected. The 'Protocol' list on the left includes Bluetooth, BPv7, DCCP, Ethernet, FC, FDDI, IEEE 802.11, IEEE 802.15.4, IPv4, IPv6, IPX, and JXTA. The main table displays traffic statistics for various endpoints, including MAC addresses and their associated traffic volumes.

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
01:00:5e:00:00:16	3	162 bytes	0	0 bytes	3	162 bytes
01:00:5e:00:00:fb	3	356 bytes	0	0 bytes	3	356 bytes
01:00:5e:00:00:fc	1	75 bytes	0	0 bytes	1	75 bytes
01:00:5e:7fff:fa	158	48 kB	0	0 bytes	158	48 kB
33:33:00:00:00:01	4	632 bytes	0	0 bytes	4	632 bytes
33:33:00:00:00:0c	7	5 kB	0	0 bytes	7	5 kB
33:33:00:00:00:16	3	270 bytes	0	0 bytes	3	270 bytes
33:33:00:00:00:fb	2	296 bytes	0	0 bytes	2	296 bytes
33:33:00:01:00:03	1	95 bytes	0	0 bytes	1	95 bytes
78:dc:b8:93:15:7a	10,996	5 MB	4,864	2 MB	6,132	3 MB
a6:c3:37:cc:6:64	10,823	5 MB	6,137	3 MB	4,686	2 MB
ff:ff:ff:ff:ff:ff	1	243 bytes	0	0 bytes	1	243 bytes

3. All TCP connections between the captured IP addresses.

The screenshot shows the 'Conversations - Wi-Fi 2' window in Wireshark. The 'Conversation Settings' panel on the left has 'Name resolution' and 'Limit to display filter' checked. The 'Copy' button is selected. The 'Protocol' list on the left includes Bluetooth, BPv7, DCCP, Ethernet, FC, FDDI, IEEE 802.11, IEEE 802.15.4, IPv4, IPv6, IPX, and JXTA. The main table displays traffic statistics for various conversations, including IP addresses and their associated traffic volumes.

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Star
172.20.10.2	63129	152.15.38.60	443	7	402 bytes	53	7	100.00%	4	228 bytes	3	174 bytes	200.72644
2607:7700:302:1436:ea0	443	2607:fb90:7515:438df1b3:2c1e3a37:ccba	63086	1	74 bytes	33	1	100.00%	1	74 bytes	0	0 bytes	63.378151
2607:7700:302:1436:ea0	443	2607:fb90:7515:438df1b3:2c1e3a37:ccba	63087	1	74 bytes	35	1	100.00%	1	74 bytes	0	0 bytes	64.413377
2607:7700:302:3de:ea5	443	2607:fb90:7515:438df1b3:2c1e3a37:ccba	63074	7	581 bytes	3	7	100.00%	5	433 bytes	2	148 bytes	0.128289
2607:7700:302:d5b6:b09	443	2607:fb90:7515:438df1b3:2c1e3a37:ccba	63050	7	595 bytes	4	7	100.00%	4	373 bytes	3	222 bytes	0.310556
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63113	2001:4706:7cd:1337	23	15	2 kB	34	15	100.00%	8	632 bytes	7	2 kB	63.875015
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63161	2001:4706:7cd:1337	23	146	12 kB	86	146	100.00%	86	6 kB	60	6 kB	363.70330
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63107	2001:489a:3403:5e7	443	30	9 kB	23	30	100.00%	14	2 kB	16	7 kB	36.170921
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63166	2600:1402:b800:36:172f:cc90	443	19	4 kB	91	19	100.00%	9	3 kB	10	1 kB	402.641741
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63051	2600:1402:b800:36:172f:cc93	443	2	148 bytes	12	2	100.00%	1	74 bytes	1	74 bytes	28.533367
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63052	2600:1402:b800:36:172f:cc93	443	2	148 bytes	14	2	100.00%	1	74 bytes	1	74 bytes	28.533775
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63101	2600:1402:b800:36:172f:cc93	443	270	162 kB	15	270	100.00%	139	119 kB	131	42 kB	28.535221
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63147	2600:1402:b800:36:172f:cc93	443	155	97 kB	72	155	100.00%	84	74 kB	71	24 kB	331.55946
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63148	2600:1402:b800:36:172f:cc93	443	24	7 kB	73	24	100.00%	12	1 kB	12	5 kB	331.65766
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63055	2600:1402:b800:36:172f:cc99	443	2	148 bytes	13	2	100.00%	1	74 bytes	1	74 bytes	28.533586
2607:fb90:7515:438df1b3:2c1e3a37:ccba	62998	2600:9000:252f:5a00:b793:ea680:93a1	443	13	1 kB	2	13	100.00%	6	448 bytes	7	605 bytes	0.115908
2607:fb90:7515:438df1b3:2c1e3a37:ccba	62990	2603:1030:a07a:400	443	6	980 bytes	47	6	100.00%	4	494 bytes	2	486 bytes	165.809921
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63025	2603:1030:a07a:400	443	3	441 bytes	65	3	100.00%	2	247 bytes	1	194 bytes	289.281581
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63057	2603:1036:303:3802:2	443	3	222 bytes	16	3	100.00%	2	148 bytes	1	74 bytes	29.056223
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63102	2603:1036:303:3802:2	443	61	50 kB	17	61	100.00%	23	5 kB	38	46 kB	29.056649
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63103	2603:1036:303:3802:2	443	18	2 kB	18	18	100.00%	11	2 kB	7	791 bytes	29.136616
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63149	2603:1036:303:3802:2	443	21	5 kB	74	21	100.00%	12	3 kB	9	2 kB	331.67408
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63154	2603:1063:27:2:254	443	32	11 kB	79	32	100.00%	13	2 kB	19	9 kB	336.568121
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63042	2607:7700:302:129b:173	443	8	444 bytes	26	8	100.00%	3	223 bytes	5	421 bytes	37.421087
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63046	2607:7700:302:12cd:a6b5	443	25	2 kB	29	25	100.00%	11	823 bytes	14	1 kB	39.203071
2607:fb90:7515:438df1b3:2c1e3a37:ccba	63130	2607:7700:302:12d1:8507	443	43	14 kB	54	43	100.00%	21	5 kB	22	9 kB	206.030311