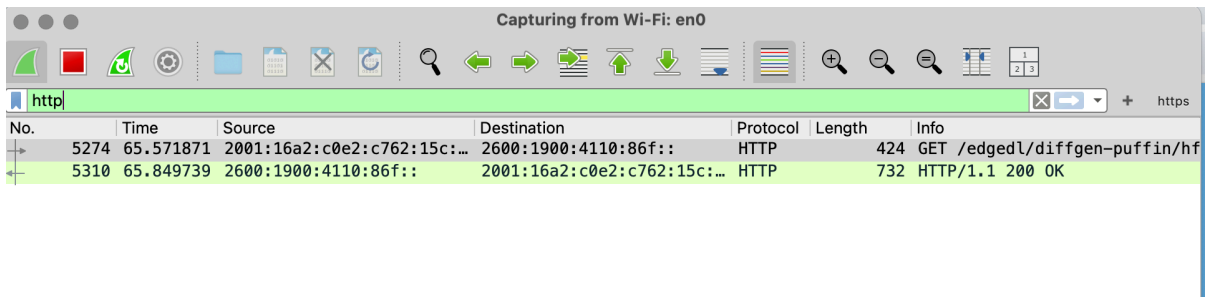


Wire Shark

Capturing HTTP Traffic.

Task 1: Start Wireshark and capture packets.

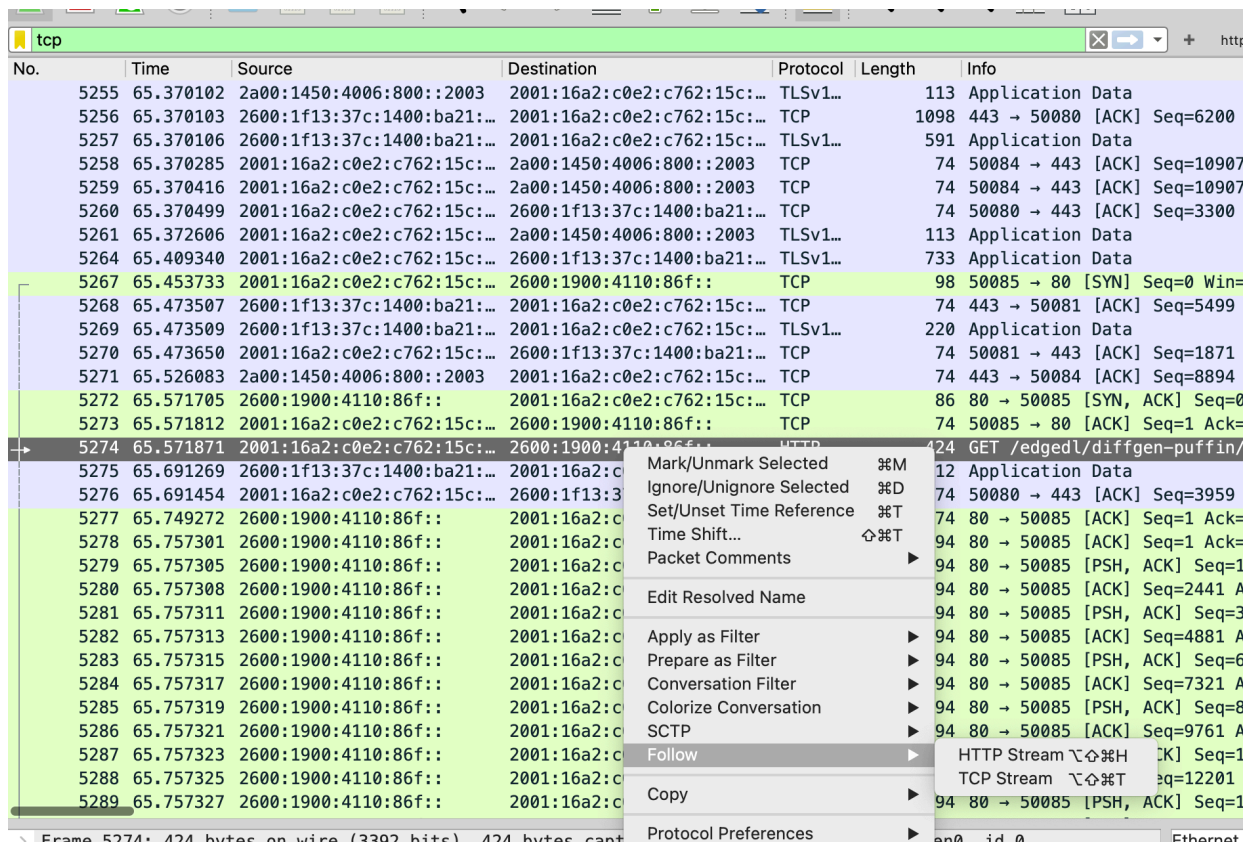
Task 2: Filter HTTP packets and analyze them.



No.	Time	Source	Destination	Protocol	Length	Info
5274	65.571871	2001:16a2:c0e2:c762:15c:...	2600:1900:4110:86f::	HTTP	424	GET /edgedl/diffgen-puffin/hf
5310	65.849739	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	HTTP	732	HTTP/1.1 200 OK

Part 2: Analyzing TCP/IP Traffic.

Task 1: Filter TCP packets.



No.	Time	Source	Destination	Protocol	Length	Info
5255	65.370102	2a00:1450:4006:800::2003	2001:16a2:c0e2:c762:15c:...	TLSv1...	113	Application Data
5256	65.370103	2600:1f13:37c:1400:ba21:...	2001:16a2:c0e2:c762:15c:...	TCP	1098	443 → 50080 [ACK] Seq=6200
5257	65.370106	2600:1f13:37c:1400:ba21:...	2001:16a2:c0e2:c762:15c:...	TLSv1...	591	Application Data
5258	65.370285	2001:16a2:c0e2:c762:15c:...	2a00:1450:4006:800::2003	TCP	74	50084 → 443 [ACK] Seq=10907
5259	65.370416	2001:16a2:c0e2:c762:15c:...	2a00:1450:4006:800::2003	TCP	74	50084 → 443 [ACK] Seq=10907
5260	65.370499	2001:16a2:c0e2:c762:15c:...	2600:1f13:37c:1400:ba21:...	TCP	74	50080 → 443 [ACK] Seq=3300
5261	65.372606	2001:16a2:c0e2:c762:15c:...	2a00:1450:4006:800::2003	TLSv1...	113	Application Data
5264	65.409340	2001:16a2:c0e2:c762:15c:...	2600:1f13:37c:1400:ba21:...	TLSv1...	733	Application Data
5267	65.453733	2001:16a2:c0e2:c762:15c:...	2600:1900:4110:86f::	TCP	98	50085 → 80 [SYN] Seq=0 Win=
5268	65.473507	2600:1f13:37c:1400:ba21:...	2001:16a2:c0e2:c762:15c:...	TCP	74	443 → 50081 [ACK] Seq=5499
5269	65.473509	2600:1f13:37c:1400:ba21:...	2001:16a2:c0e2:c762:15c:...	TLSv1...	220	Application Data
5270	65.473650	2001:16a2:c0e2:c762:15c:...	2600:1f13:37c:1400:ba21:...	TCP	74	50081 → 443 [ACK] Seq=1871
5271	65.526083	2a00:1450:4006:800::2003	2001:16a2:c0e2:c762:15c:...	TCP	74	443 → 50084 [ACK] Seq=8894
5272	65.571705	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	86	80 → 50085 [SYN, ACK] Seq=0
5273	65.571812	2001:16a2:c0e2:c762:15c:...	2600:1900:4110:86f::	TCP	74	50085 → 80 [ACK] Seq=1 Ack=
5274	65.571871	2001:16a2:c0e2:c762:15c:...	2600:1900:4110:86f::	HTTP	424	GET /edgedl/diffgen-puffin/
5275	65.691269	2600:1f13:37c:1400:ba21:...	2001:16a2:c0e2:c762:15c:...	TCP	12	Application Data
5276	65.691454	2001:16a2:c0e2:c762:15c:...	2600:1f13:37c:1400:ba21:...	TCP	74	50080 → 443 [ACK] Seq=3959
5277	65.749272	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	74	80 → 50085 [ACK] Seq=1 Ack=
5278	65.757301	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [ACK] Seq=1 Ack=
5279	65.757305	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [PSH, ACK] Seq=1
5280	65.757308	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [ACK] Seq=2441 A
5281	65.757311	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [PSH, ACK] Seq=3
5282	65.757313	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [ACK] Seq=4881 A
5283	65.757315	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [PSH, ACK] Seq=6
5284	65.757317	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [ACK] Seq=7321 A
5285	65.757319	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [PSH, ACK] Seq=8
5286	65.757321	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [ACK] Seq=9761 A
5287	65.757323	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [ACK] Seq=1
5288	65.757325	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [PSH, ACK] Seq=1
5289	65.757327	2600:1900:4110:86f::	2001:16a2:c0e2:c762:15c:...	TCP	94	80 → 50085 [PSH, ACK] Seq=1

Task 2: Analyze TCP handshake and investigate Data Transfer and Termination.

Wireshark - Follow TCP Stream (tcp.stream eq 151) - Wi-Fi: en0

No.	Time	Source	Destination	Protocol	Length	Info
5715	102.3...	2001:16a2:c0e2:c...	2600:1f13:37c:14...	TCP	98	59525 → 80 [SYN] Seq=0 Win=65535 Len=0 M
5744	102.6...	2600:1f13:37c:14...	2001:16a2:c0e2:c...	TCP	86	80 → 59525 [SYN, ACK] Seq=0 Ack=1 Win=26
5745	102.6...	2001:16a2:c0e2:c...	2600:1f13:37c:14...	TCP	74	59525 → 80 [ACK] Seq=1 Ack=1 Win=262144
5788	103.3...	2001:16a2:c0e2:c...	2600:1f13:37c:14...	HTTP	464	GET /favicon.ico HTTP/1.1
5800	103.6...	2600:1f13:37c:14...	2001:16a2:c0e2:c...	TCP	74	80 → 59525 [ACK] Seq=1 Ack=391 Win=28032
5801	103.6...	2600:1f13:37c:14...	2001:16a2:c0e2:c...	HTTP	517	HTTP/1.1 200 OK (PNG)
5802	103.6...	2001:16a2:c0e2:c...	2600:1f13:37c:14...	TCP	74	59525 → 80 [ACK] Seq=391 Ack=444 Win=261
5917	108.6...	2600:1f13:37c:14...	2001:16a2:c0e2:c...	TCP	74	80 → 59525 [FIN, ACK] Seq=444 Ack=391 Wi
5918	108.6...	2001:16a2:c0e2:c...	2600:1f13:37c:14...	TCP	74	59525 → 80 [ACK] Seq=391 Ack=445 Win=262
5919	108.6...	2001:16a2:c0e2:c...	2600:1f13:37c:14...	TCP	74	59525 → 80 [FIN, ACK] Seq=391 Ack=445 Wi
5923	108.9...	2600:1f13:37c:14...	2001:16a2:c0e2:c...	TCP	74	80 → 59525 [ACK] Seq=445 Ack=392 Win=280

Frame 5788: 464 bytes on wire (3712 bits), 464 bytes captured on interface Wi-Fi: en0, 464 bytes from 2001:16a2:c0e2:c...:6025c:1f to 2600:1f13:37c:14...:6025c:1f, Ethernet II, Src: 5e:a2:03:60:5c:1f (5e:a2:03:60:5c:1f), Internet Protocol Version 6, Src: 2001:16a2:c0e2:c762:10e...

Host: shinyclearfreshmoon.neverssl.com
Referer: http://shinyclearfreshmoon.neverssl.com/online/
Accept: */*
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/18.0 Safari/605.1.15
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Priority: u=3, i
Accept-Encoding: gzip, deflate
Connection: keep-alive

Part 3: Capturing and Analyzing UDP Traffic

Task 1: Generate UDP traffic and capture packets

Task 2: Filter and analysis UDP Packets

Wireshark - udp

No.	Time	Source	Destination	Protocol	Length	Info
33	3.081...	10.90.18.178	10.90.255.255	NBNS	92	Name query NB TMOSCE<00>
34	3.118...	10.90.46.5	172.224.169.8	UDP	83	55428 → 443 Len=41
36	3.219...	172.224.169.8	10.90.46.5	UDP	74	443 → 55428 Len=32
37	3.261...	10.90.4.151	10.90.255.255	NBNS	92	Name query NB BRWDCA2662DE5A0<00>
38	3.370...	10.90.15.126	10.90.255.255	NBNS	92	Name query NB ISATAP<00>
39	3.504...	10.90.46.5	172.224.169.8	UDP	83	55428 → 443 Len=41
40	3.515...	10.90.44.129	230.0.0.1	UDP	92	63993 → 6666 Len=50
41	3.609...	172.224.169.8	10.90.46.5	UDP	73	443 → 55428 Len=31
42	3.772...	10.90.4.151	10.90.255.255	NBNS	92	Name query NB BRWDCA2662DE5A0<00>
43	4.991...	10.90.46.5	172.224.169.8	UDP	1279	55428 → 443 Len=1237
52	5.105...	172.224.169.8	10.90.46.5	UDP	74	443 → 55428 Len=32
61	5.208...	10.90.4.151	10.90.255.255	NBNS	92	Name query NB BRWDCA2662DE5A0<00>
65	5.232...	172.224.169.8	10.90.46.5	UDP	498	443 → 55428 Len=456
66	5.234...	10.90.46.5	172.224.169.8	UDP	99	55428 → 443 Len=57
70	5.332...	172.224.169.8	10.90.46.5	UDP	74	443 → 55428 Len=32
88	5.518...	10.90.77.84	10.90.255.255	NBNS	92	Name query NB MACB00KAIR-B12C<00>
89	5.528...	10.90.44.129	230.0.0.1	UDP	92	63993 → 6666 Len=50
90	5.614...	10.90.46.5	172.224.169.8	UDP	83	55428 → 443 Len=41
93	5.723...	172.224.169.8	10.90.46.5	UDP	74	443 → 55428 Len=32
94	5.724...	10.90.46.5	172.224.169.8	UDP	87	55428 → 443 Len=45
95	5.844...	172.224.169.8	10.90.46.5	UDP	73	443 → 55428 Len=31
122	6.450...	10.90.32.169	10.90.255.255	UDP	70	50130 → 22222 Len=28
123	6.458...	10.90.32.169	10.90.255.255	UDP	60	50133 → 3289 Len=14
141	6.646...	10.90.15.171	10.90.255.255	NBNS	92	Name query NB WORKGROUP<1c>
142	6.652...	10.90.4.151	10.90.255.255	NBNS	92	Name query NB BRWDCA2662DE5A0<00>
164	6.984...	10.90.77.84	10.90.255.255	NBNS	92	Name query NB MACB00KAIR-B12C<00>

Step 4: Observe the source and destination ports, length, and data

24	0.984...	10.90.46.5	172.224.169.8	UDP	83	55428 → 443	Len=41
25	1.117...	172.224.169.8	10.90.46.5	UDP	74	443 → 55428	Len=32
26	1.308...	10.90.46.5	172.224.169.8	UDP	83	55428 → 443	Len=41
27	1.499...	172.224.169.8	10.90.46.5	UDP	73	443 → 55428	Len=31
28	1.534...	10.90.1.208	255.255.255.255	UDP	60	63299 → 8610	Len=16
29	1.536...	10.90.1.208	255.255.255.255	UDP	60	63299 → 8610	Len=16
30	1.536...	10.90.1.208	10.90.255.255	NPMS	62	Name query NP	TMOSCE-00...

Destination Address: 10.90.46.5
[Stream index: 5]

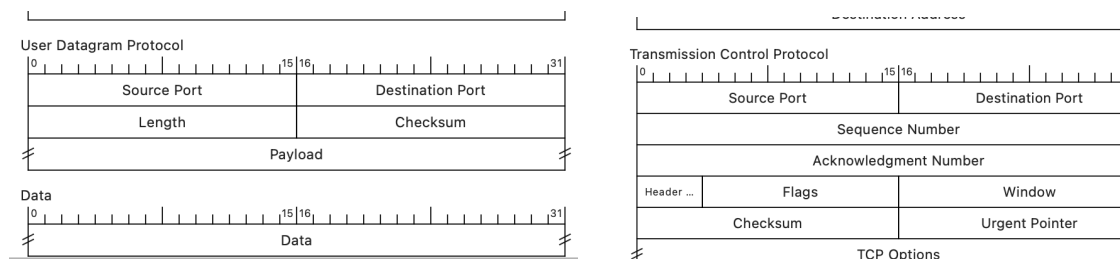
✓ User Datagram Protocol, Src Port: 443, Dst Port: 55428

Source Port: 443
Destination Port: 55428
Length: 40
Checksum: 0xbc73 [unverified]
[Checksum Status: Unverified]
[Stream index: 5]
[Stream Packet Number: 2]
[Timestamps]
UDP payload (32 bytes)

✓ Data (32 bytes)
Data: 51530eed6b9cc4426ae92ccd9876ca564b0264c5009d3079422d814106b4a4fe
[Length: 32]

Step 5: Compare the simplicity of UDP headers with TCP headers.

	UDP	TCP
Header Size	8 bytes	20-60 bytes



Part 4: Comparing TCP and UDP by filling in the following tables. Save your work (e.g., in an MS Word document)

Task 1: Fill in the following table and provide reasons.

Feature	TCP or UDP	Reasons
Reliability and Connection Establishment	TCP	TCP is a connection-oriented protocol that ensures reliable communication. It establishes a connection using a three-way handshake

Data Integrity and Ordering	TCP	TCP provides error checking, retransmission, and ordering of packets to ensure data integrity. It ensures that data is received in the correct sequence. UDP, does not guarantee ordering or delivery.
------------------------------------	-----	--

Task 2: Identify the use Cases and Performance of TCP and UDP.

Feature	TCP	UDP
Use Cases	Web browsing (HTTP/HTTPS), email (SMTP, IMAP, POP3), file transfer (FTP)	Real-time applications like VoIP, video streaming, online gaming, and DNS
Performance	Slower due to connection establishment, error checking, and retransmissions	Faster because it is connectionless, has low overhead, and does not wait for acknowledgments or retransmit lost packets.