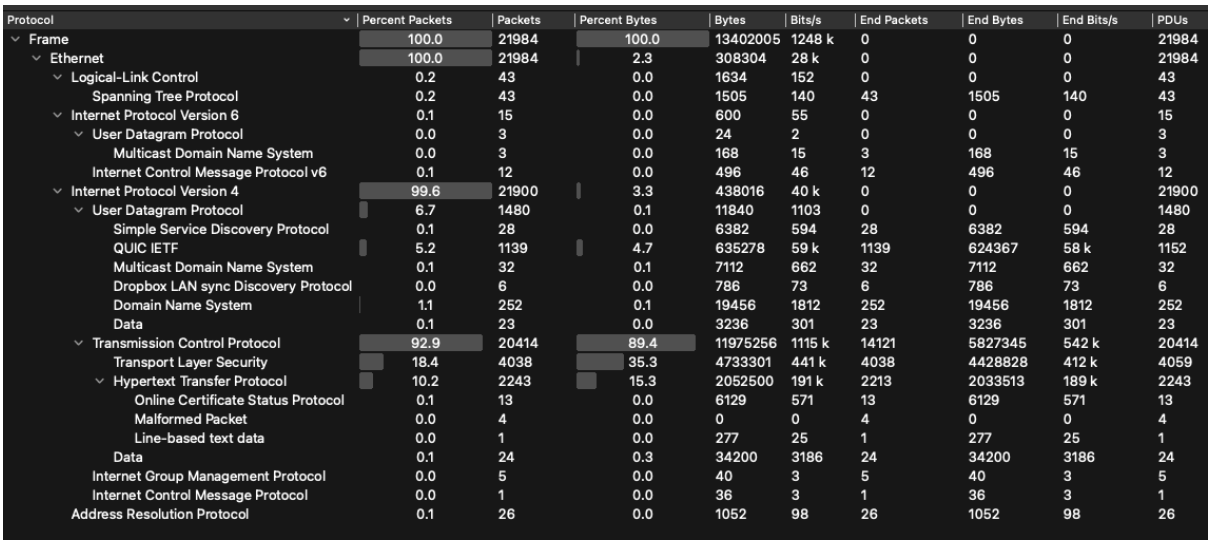
Question 1:



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	21984	100.0	13402005	1248 k	0	0	0	21984
Ethernet	100.0	21984	2.3	308304	28 k	0	0	0	21984
Logical-Link Control	0.2	43	0.0	1634	152	0	0	0	43
Spanning Tree Protocol	0.2	43	0.0	1505	140	43	1505	140	43
Internet Protocol Version 6	0.1	15	0.0	600	55	0	0	0	15
User Datagram Protocol	0.0	3	0.0	24	2	0	0	0	3
Multicast Domain Name System	0.0	3	0.0	168	15	3	168	15	3
Internet Control Message Protocol v6	0.1	12	0.0	496	46	12	496	46	12
Internet Protocol Version 4	99.6	21900	3.3	438016	40 k	0	0	0	21900
User Datagram Protocol	6.7	1480	0.1	11840	1103	0	0	0	1480
Simple Service Discovery Protocol	0.1	28	0.0	6382	594	28	6382	594	28
QUIC IETF	5.2	1139	4.7	635278	59 k	1139	624367	58 k	1152
Multicast Domain Name System	0.1	32	0.1	7112	662	32	7112	662	32
Dropbox LAN sync Discovery Protocol	0.0	6	0.0	786	73	6	786	73	6
Domain Name System	1.1	252	0.1	19456	1812	252	19456	1812	252
Data	0.1	23	0.0	3236	301	23	3236	301	23
Transmission Control Protocol	92.9	20414	89.4	11975256	1115 k	14121	5827345	542 k	20414
Transport Layer Security	18.4	4038	35.3	4733301	441 k	4038	4428828	412 k	4059
Hypertext Transfer Protocol	10.2	2243	15.3	2052500	191 k	2213	2033513	189 k	2243
Online Certificate Status Protocol	0.1	13	0.0	6129	571	13	6129	571	13
Malformed Packet	0.0	4	0.0	0	0	4	0	0	4
Line-based text data	0.0	1	0.0	277	25	1	277	25	1
Data	0.1	24	0.3	34200	3186	24	34200	3186	24
Internet Group Management Protocol	0.0	5	0.0	40	3	5	40	3	5
Internet Control Message Protocol	0.0	1	0.0	36	3	1	36	3	1
Address Resolution Protocol	0.1	26	0.0	1052	98	26	1052	98	26

The picture above was taken from Wireshark by going to **Statistics -> Protocol hierarchy.** It summarizes all the protocols that were found in the file during the capturing in hierarchical order.

The list of protocols is as follows:

ARP: a layer 2 protocol used to map MAC addresses to IP addresses.

DB-LSP-DISC/JSON: allows computers on a local network to synchronize files across other computers.

DNS: translates human readable domain names to machine readable IP addresses

HTTP: is the set of rules for transferring files such as text, images, sound, video and other multimedia files over the web.

ICMP: is a network layer protocol used by network devices to diagnose network communication issues.

ICMPv6: is the implementation of the ICMP for Internet Protocol version 6 (IPv6).

IGMPv2: is a protocol that allows several devices to share one IP address so they can all receive the same data. It is the revised version of IGMPv1 communication protocol.

MDNS: is a protocol that resolves hostnames to IP addresses within small networks that do not include a local name server.

OCSP: is a protocol that maintains the security of a server and other network resources.

QUIC: is an experimental general-purpose transport layer network protocol designed to reduce latency compared to that of TCP.

SSDP: is a protocol that discovers what devices (and their capabilities) are available in a local area network.

STP: is a Layer 2 network protocol used to prevent looping within a network topology.

TCP: is a transport protocol that is used on top of IP to ensure reliable transmission of packets.

TLSv1.2: is the successor to Secure Sockets Layer (SSL) used by endpoint devices and applications to authenticate and encrypt data securely when transferred over a network.

TLSv1.3: a major revision to the TLS protocol that is intended to provide better security and improve handshake performance.

UDP: is a communications protocol that is primarily used to establish low-latency and loss-tolerating connections between applications on the internet.

Question 2:

www.upei.ca → 20.48.225.96
pki-goog.l.google.com → 142.250.80.67
ocsp.digicert.com → 72.21.91.29
ocsp.sca1b.amazontrust.com → 18.164.93.157
proxy-safebrowsing.googleapis.com → 17.253.3.213
updates-http.cdn-apple.com → 17.253.3.195

Question 3:

adservice.google.com
analytics.tiktok.com
apis.google.com
cdn.smoot.apple.com
cdn2.smoot.apple.com
cdnjs.cloudflare.com
clients1.google.com
connect.facebook.net
cse.google.com

d.dropbox.com
e673.dsce9.akamaiedge.net
en.wikipedia.org
files.upei.ca
fonts.gstatic.com
googleads.g.doubleclick.net
id.google.com
in.hotjar.com
intake-analytics.wikimedia.org
login.wikimedia.org
mask-api.icloud.com
mask.icloud.com
mesu-cdn.origin-apple.com.akadns.net
meta.wikimedia.org
ocsp.pki.goog
ocsp.sca1b.amazontrust.com
p.typekit.net
pixel.sitescout.com
pixel.tapad.com
play.google.com
proxy.safebrowsing.apple
safebrowsing.googleapis.com
sc-static.net
script.hotjar.com
static.hotjar.com
stats.g.doubleclick.net
themes.googleusercontent.com
token.safebrowsing.apple
tr.snapchat.com
up.pixel.ad
updates-http.cdn-apple.com
upload.wikimedia.org
use.typekit.com
vars.hotjar.com
www.facebook.com
www.google-analytics.com
www.google.ca

www.google.com
www.googleadservices.com
www.googleapis.com
www.googletagmanager.com
www.gstatic.com
www.upei.ca

Question 4:

The only interesting finding I found while analyzing the file is that we have 23 unrecognized packets under the UDP protocol and 24 unrecognized packets under the TCP protocol with the name “Data.” From what I found online, having “data” in our file is not good because Wireshark does not recognize the application running under TCP and UDP.