Project 3-An Experimental Investigation into Gmail and Outlook

SCICOMP202 Networks and Communications

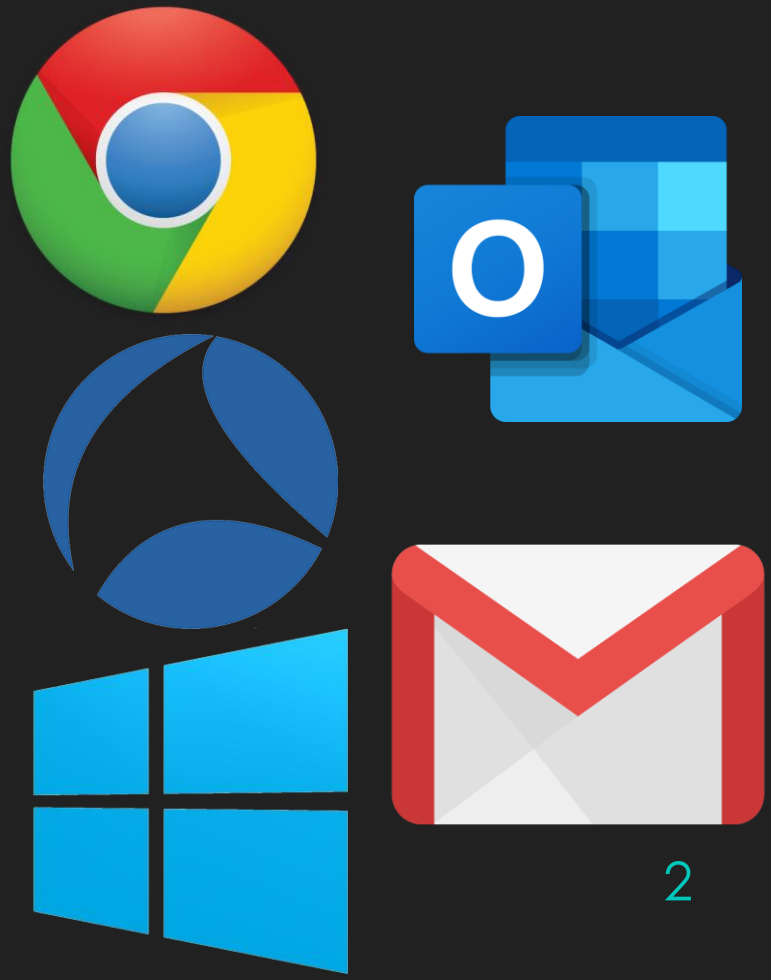
Joanikij Chulev

This is all my own work. I have not knowingly allowed others to copy my work. This work has not been submitted for assessment in any other context.

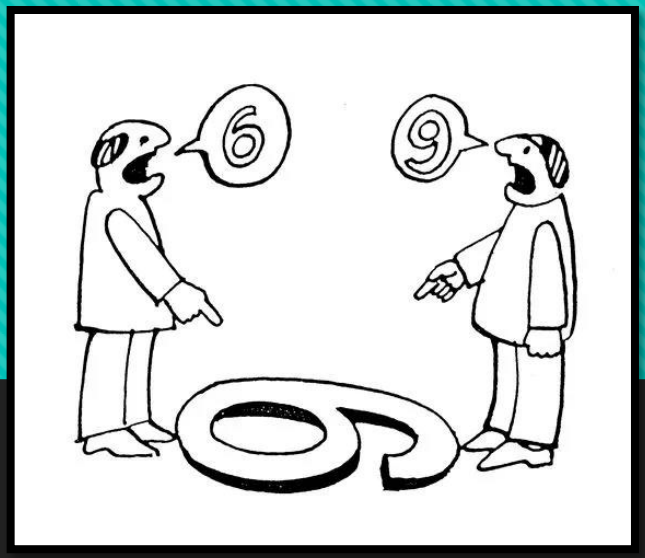
Tracking information

When gathering the data I closed all tabs, cleared recent cache and turned off background application services that may be using internet.

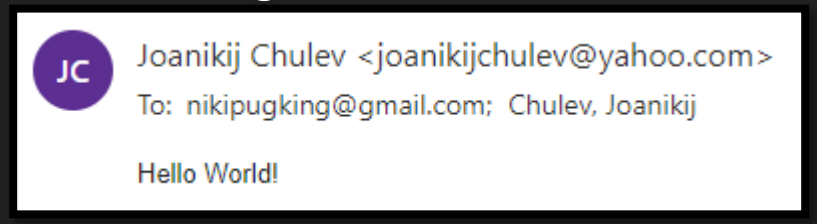
- Processor: Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz
- Installed RAM:8.00 GB (7.80 GB usable)
- System type: 64-bit operating system, x64-based processor
- Tracing was done in Wireshark.
- Wireshark version: 3.6.2 (v3.6.2-0-g626020d9b3c3)
- Connection type: Wireless(Wi-fi)
- SSID: eduroam
- Browsers used: Chrome [Version 98.0.4758.102 (64-bit)]
- Email sites used: Gmail and Outlook
- **When getting the statistics the ip.addr== filter was set to my ip addres!**

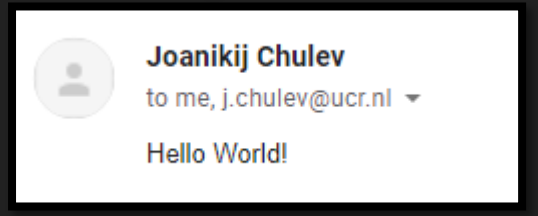


Assumptions made and info



- This study is based on which packets are used and packet traffic prominent in opening a mail from these sites. I assumed that opening a mail from these sites, whether Outlook or Gmail does not depend on content withing the mails, due to the fact that we are not opening or downloading any content only the secured mail messages.
- It was also assumed that the content type does not matter, so a simple "Hello World!" text string was sent to both of the mails, from a separate yahoo mail.
- It was assumed that no caching of any sorts in regards of these sent mails has been saved or used when doing the track, which would change the results in the track drastically.





Hypotheses and expectations

Main Hypotheses:

- Null Hypothesis (H0): The packets used when opening mail in both scenarios are NOT the same type of packets.
- Alternative Hypothesis (H1): The packets used when opening mail in both scenarios are the same type of packets.

Secondary Hypotheses:

- Null Hypothesis (H0): The packets used when opening mail in both scenarios do NOT use conventional packets (UDP+QUIC/TCP).
- Alternative Hypothesis (H1): The packets used when opening mail in both scenarios do use conventional packets (UDP+QUIC/TCP).

Gmail study

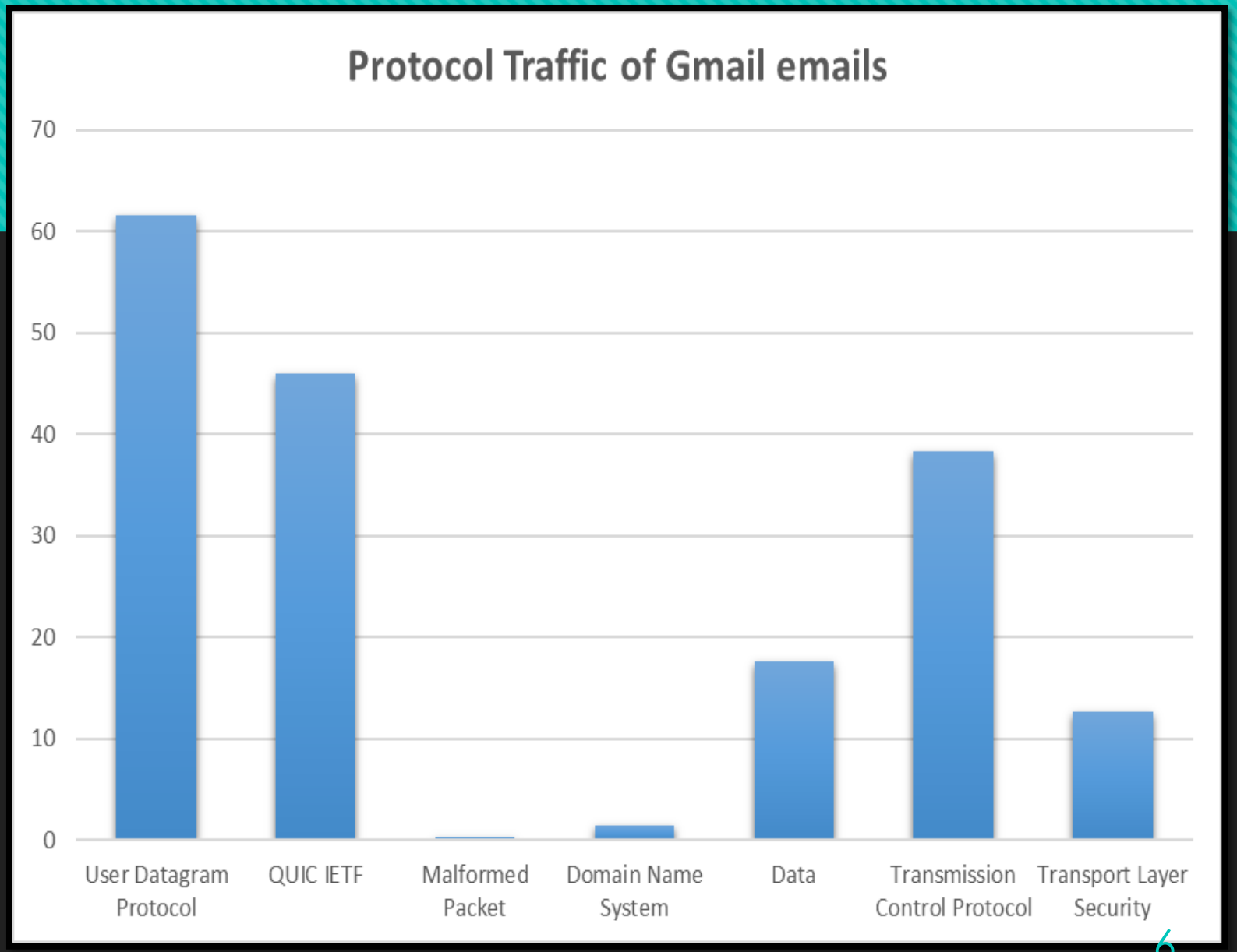
- As we can see from the figure a total of 396 packets were sent. Of which 244 are UDP packets, more specifically most are QUIC packets, using the google developed protocol which google seems to like to use. This was not a surprise. (70 of the UDP packet are data packets). In contrast 152 packets were TCP packets, and only 50 were TLS packets.
- The total size in bytes of all packets was 112627 bytes.

***We also got 1 malformed packet. Malformed packet means that the protocol dissector can't dissect the contents of the packet any further.**

Protocol	Packets	Bytes
Frame	396	112627
Ethernet	396	5544
Internet Protocol Version 4	396	7920
User Datagram Protocol	244	1952
QUIC IETF	182	59399
Malformed Packet	1	0
Domain Name System	6	347
Data	70	9523
Transmission Control Protocol	152	30320
Transport Layer Security	50	27048

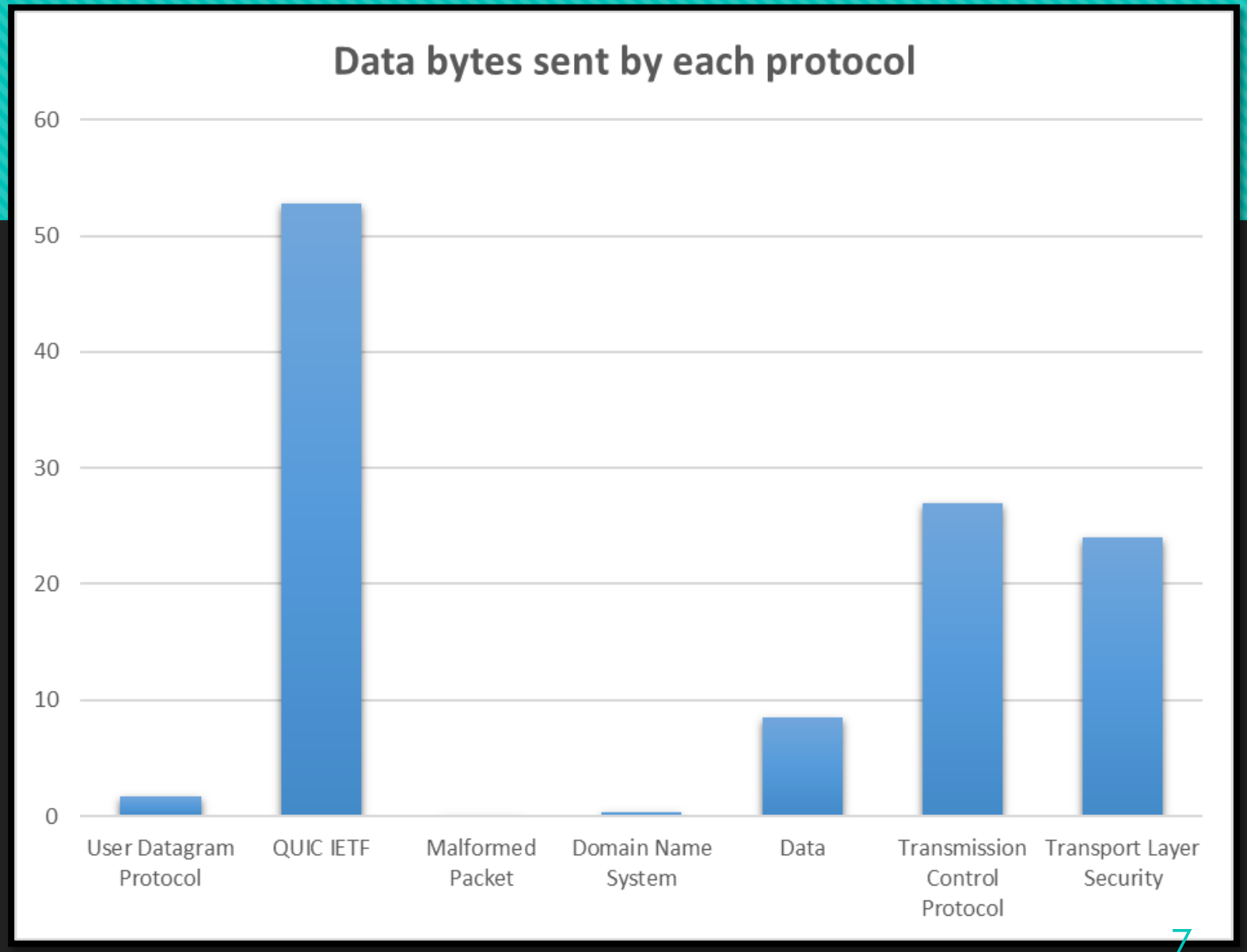
Gmail traffic

- As we can see by this figure a total of 61.62% of all packets were UDP packets, with QUIC taking up 45.95% of the packets, and DNS packets were a mere 1.52%. 17.67% of the packets were data UDP packets.
- In contrast 38.38% of the packets were TCP, of which 12.63% were TLS packets.



Gmail data

- As we can see by this figure, 1.73% of all data sent in bytes is taken up by UDP. 52.74% of all data sent was used up by QUIC. DNS took up only 0.3% of the data. And the data UDP packets took up 8.4% of the data sent in bytes.
- In contrast 26.92% of the data bytes were TCP. TLS taking up 24.02% of the bytes.



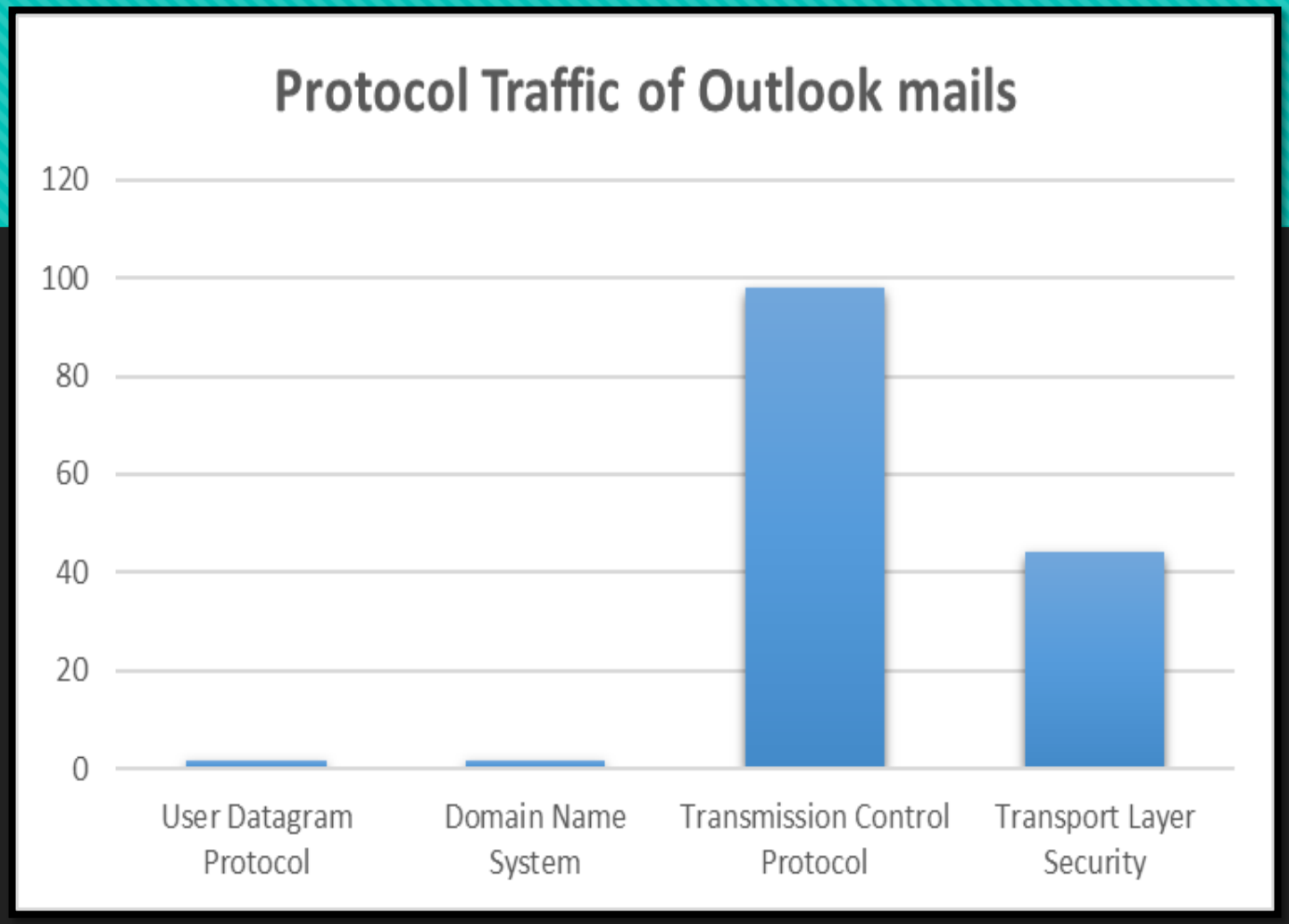
Outlook study

- As we can see from the figure a total of 749 packets were sent. Of which only 14 are UDP packets, more specifically DNS packets to resolve domains, so no actual data of the mail was sent in UDP. In contrast 735 packets were TCP packets, and 331were TLS packets.
- The total size in bytes of all packets was 948306 bytes.

Protocol	Packets	Bytes
Frame	749	948306
Ethernet	749	10486
Internet Protocol Version 4	749	14980
User Datagram Protocol	14	112
Domain Name System	14	1631
Transmission Control Protocol	735	920863
Transport Layer Security	331	960202

Outlook traffic

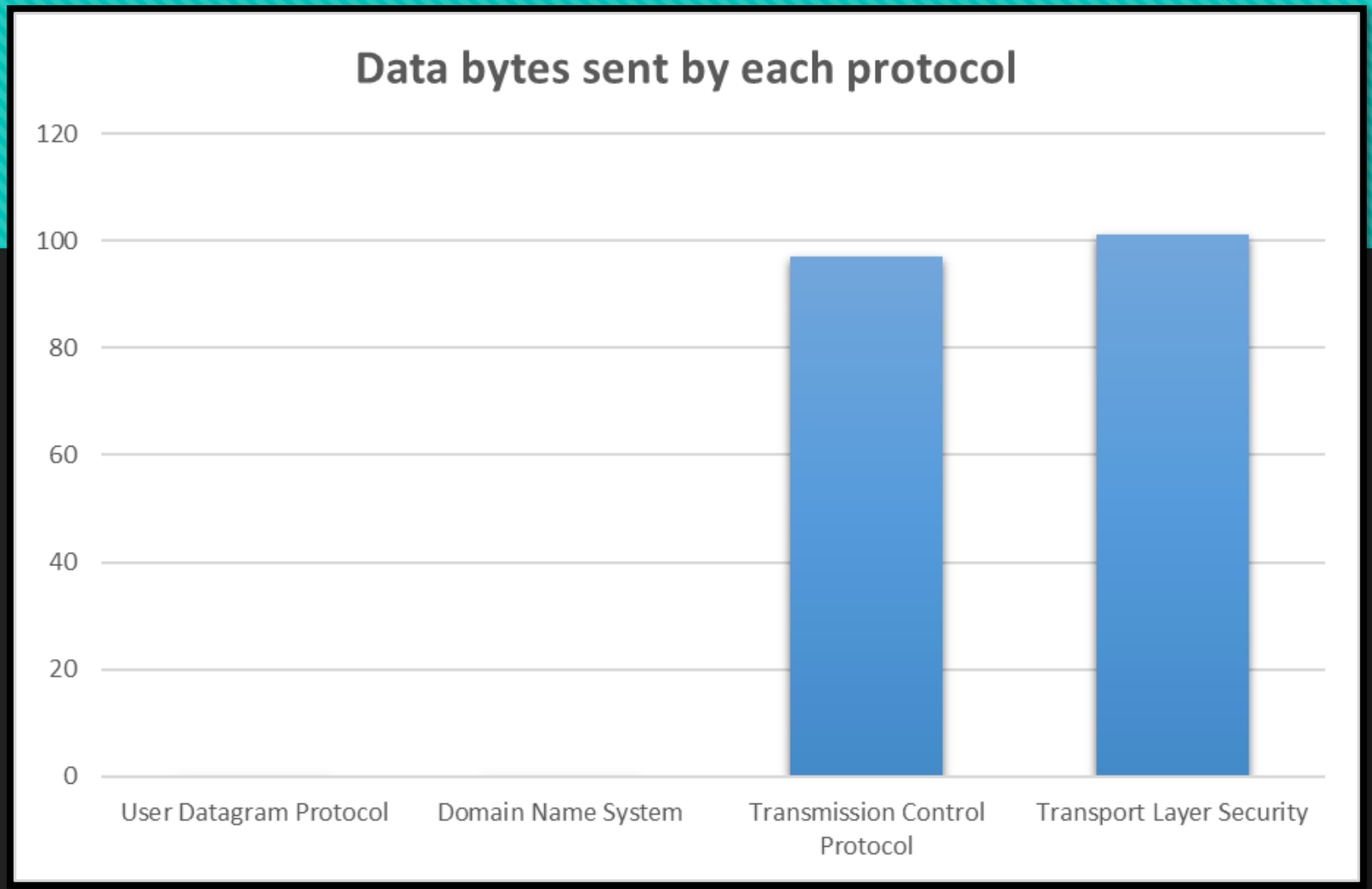
- As we can see by this figure a total of 1.87% of all packets were UDP packets, with DNS taking up 1.87% of the packets, meaning all UDP were DNS packets.
- In contrast 98.13% of the packets were TCP, of which 44.19% were TLS packets.



Outlook data

- As we can see by this figure, 0.01% of all data sent in bytes is taken up by UDP. DNS took up only 0.17 % of the data..
- In contrast 97.1% of the data bytes was TCP. TLS taking up 101.2% of the bytes.

***One explanation to the number going over 100% in regards of the TLS packets is that some packets may not be recorded in the total number of packets but the sent data in bytes is.**



Comparison of the statistics

- We can clearly see that Outlook sent way more data than Gmail, almost 9 fold. As well as the packets sent from Outlook seem to be 2 times the amount the packets sent by Gmail. Gmail seems to utilize UDP (QUIC) and Outlook seems to utilize only TCP to send data. Both use TLS for security purposes, but Outlook uses the TLS protocol almost 4 times more.

396	Packets	749
112627	Bytes	948306
61.62%	UDP in %	1.87%
38.38%	TCP in %	98.13%
12.63%	TLS in %	44.19%

Conclusion

- **Main Conclusion:** We accept the Null Hypothesis, meaning we accept H0 for this section. The packets utilized in the 2 scenarios are different in numbers, and Gmail utilizes QUIC.

- **Secondary Conclusion: :** We accept the Alternative Hypothesis, meaning we accept H1 for this section. The packets used are basic UDP and TCP based packets to handle the data being sent.
- Special mailing protocols like the Simple Mail Transfer Protocol (SMTP), the Post Office Protocol (POP), and the Internet Message Access Protocol (IMAP) were not utilized.

References

-Google images;
*https://www.wireshark.org/docs/wsug_html_chunked/AppMessages.html#:~:text=Malformed%20packet%20means%20that%20the,known%20TCP%20or%20UDP%20port.;
*https://osqa-ask.wireshark.org/questions/52190/what-is-the-reason-for-malformed-packet-error/
*https://en.wikipedia.org/wiki/Transport_Layer_Security
*https://www.wireshark.org/docs/wsug_html_chunked/ChStatHierarchy.html