Protocol Analysis of Student Internet Usage on TCU's Network

Alexa Mercado
a.mercado89@tcu.edu

Gabby Campos
g.e.campos@tcu.edu

August 11, 2025

Abstract

Over the years, there have been many studies looking at network traffic analysis; however, most of these studies do not take into account the most common protocols used while accessing websites and applications commonly used. Analyzing the most common protocols that TCU students use daily as they access different websites and applications can give us a better understanding of the security and efficiency of the websites and applications used by students. Furthermore, through this analysis, it was deduced that generally, the daily network activity of TCU students is secure and efficient.

1 Introduction

Computer networks are at the cornerstone of modern lives, they allow for the use of various services from widespread communication, web browsing, and gaming. As the number of users increases on a network, analyzing network traffic trends in the protocols used is crucial to network efficiency and security. This paper will target investigating the types of network traffic which is the amount of data moving across a computer network and network protocols that govern data communication between devices. The scope of this research will cover network traffic and protocols TCU students use while engaging in different types of network activities.

To better understand the importance analysis of network traffic, specifically protocols used by students in the network, taking a closer look at past and recent studies is crucial. Most recently, in 2017 there have been studies such as "Enhance the ICS Network Security using the Whitelist-Based Network Monitoring through Protocol Analysis." that take a look at industrial

control system (ICS) protocols commonly used in industries such as nuclear and petroleum facilities [1]. In this study, typical everyday protocols such as file transfer protocol (FTP) a protocol that handles file transfers, Modbus/TCP an industrial protocol used to give commands to devices, were used to analyze the security of a network. Through a more accurate traffic analysis, the study was able to successfully test the security of ICS protocols. The method used in the study was reverse engineering to "monitor network traffic and achieve more accurate traffic analysis." The study achieved this by taking a look at how Modbus TCP handles whitelisting commands, a system where only commands that are preapproved are allowed on a system and FTP to test generalizability, protocols commonly used in ICS systems. This is a crucial finding as ICS is commonly used in huge industries to automate commands issued to devices via a network. Furthermore, by improving ICS security, industries are better suited against cyber-attacks that could damage the network security of businesses. Additionally, the security of ICS systems is important to prevent physical losses and monetary losses. Nonetheless, keeping ICS systems and many other network systems safe will improve security and efficiency of large-scale industries and businesses.

Overall, through this study, a key finding is that network users are directly impacted by the security of a network and taking a look at protocols involved in a network can provide useful insight into network traffic and its security and efficiency. In this study, similar methods can be applied to analyze protocols involved in TCU student's daily network usage. This analysis hopes to answer the questions: What are the most common network protocols used by TCU students? How do these different types of network activities affect network traffic patterns and bandwidth usage? How secure are some of the sites that students on TCU's campus use?

2 Methodology

2.1 Proposed Methodology

In this study, analysis was conducted using Wireshark captures from many common websites and applications students commonly use on TCU's campus. Specifically, analysis was done using captures from TCU Box, Cengage, Canva, Grammarly, Wikipedia, Quizlet, TCU.edu, TCU Online, TCU Campus Rec Website, Shared Google Docs, Shared Google Slides, Google Drive, PowerPoint Online, Word Online, TCU Bookstore, ChatGPT, Google Search Engine, Purple Schedule Builder, TCU Library, Google Map, MyFitnessPal, Online Game Streaming, YouTube, Spotify, Hulu, Netflix, Reddit,

TikTok, Snapchat, Instagram, Twitch, LinkedIn, Handshake, Indeed, Discord, GitHub, Gmail, Zoom, Microsoft Teams, and Outlook.

2.2 Extracting data using Wireshark

To obtain results, Wireshark was used to access each individual website or app one at a time and capture all network packets associated with the specific website or address analyzed. The duration of each capture in Wireshark was exactly 10 seconds. To achieve this, the capture results were filtered using the filter `(frame.time_relative >= X && frame.time_relative <= X+10)`, where `X` varied depending on the capture. Each 10-second window was then saved as its own .pcap file, named after the website or service it captured.

To isolate traffic between the test device and the domain visited, IP-filtering methods were used for each 10-second capture. To begin, a dns.query.name search is performed on each capture as depicted in Figure 1

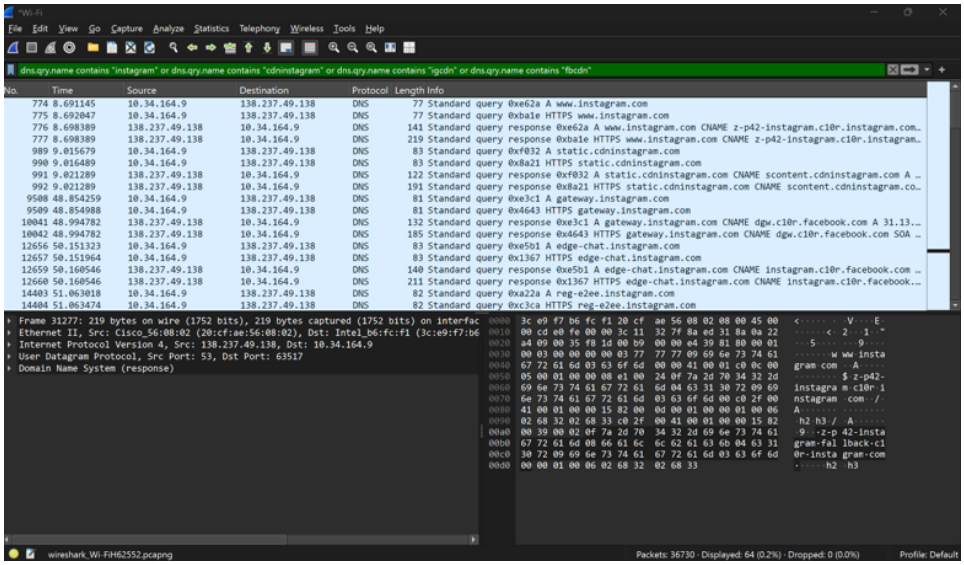

Figure 1: Filtering DNS queries using the `dns.query.name` field in Wireshark on an Instagram capture.

The example depicted in Figure 1 shows a DNS query for Instagram, which displays Instagram domains and their associated IP addresses. After applying this filter, all IP addresses for associated domains are noted, and

applied as a separate filter which isolates traffic between the device and associated IP's as shown in Figure 2.

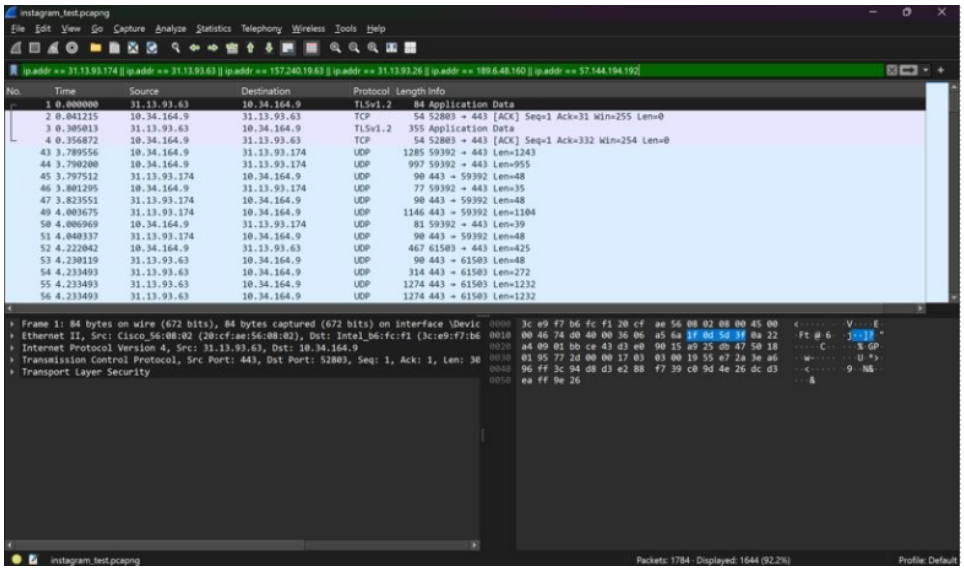

Figure 2: IP address filtering applied to Instagram capture.

To accurately analyze the captured data, the results from the Protocol Hierarchy in Wireshark were recorded. The specific data in the protocol hierarchy that were recorded were each protocol name, the percentage of packets, the number of packets, the percentage of bytes, and the number of bytes in a table using Microsoft Excel. This process was repeated for each individual website and application.

2.3 Merging Results

Once all the data for each individual website was obtained, the individual results were merged into one master table that listed every single protocol that was found according to Wireshrk's Protocol Hierarchy. The full list of protocols found was: 802.1Q Virtual LAN, Data, Domain Name System, Ethernet, Frame, HiPerConTracer Trace Service, Hypertext Transfer Protocol, Internet Control Message Protocol, Internet Protocol, Version 4, Line-based text data, Link-local Multicast Name Resolution, Malformed Packet, Media Type, QUIC IET, Real-time Transport Control Protocol, Transmission Control Protocol, Transport Layer Security, User Datagram

Protocol, WireGuard Protocol. Redundant and irrelevant protocols were removed from the final analysis.

After removing irrelevant protocols listed by Wireshark's Protocol Hierarchy, a final table was created that included all relevant protocols. Below is a list of all relevant protocols that were captured and analyzed in this study:

Domain Name System (DNS) is a protocol that is used to convert human-readable domain names into IP addresses (i.e. 138.237.49.138 is converted into instagram.com) [3]. **Transport Layer Security** (TLS) is a protocol that is used to ensure secure communications over a network between devices [4]. **Quick User Datagram Protocol Internet Connections as used by the Internet Engineering Task Force** (QUIC IETF) is a protocol that is used to make communication faster and more secure, and is used primarily for things such as web browsing, video streaming, and real-time communication [5]. **Transmission Control Protocol** (TCP) is a protocol used to ensure reliable data delivery between applications, and consists of various control packets [6]. **User Datagram Protocol** (UDP) is a lightweight protocol that ensures quick data transmission, and is used primarily for streaming in real time for applications such as video games or real-time communication platforms [7]. Finally, **Hypertext Transfer Protocol** (HTTP) is a protocol that allows for the transfer of data between a client and a web server [8].

These protocols, the number of packets, number of bytes, the percent of packets, and the percent of bytes were all combined into one table. To understand this process please see Appendix 5.

3 Data Presentation

After all data was combined into one table in Excel, the data was visualized in the bar chart shown in Figure 3. All protocols analyzed are visualized along with the percentage of byte packets, and percentage of bytes as collected across all sites. This visualization highlights the frequency of each protocol in terms of packets and bytes. This data can explain how student network usage varies across the campus network.

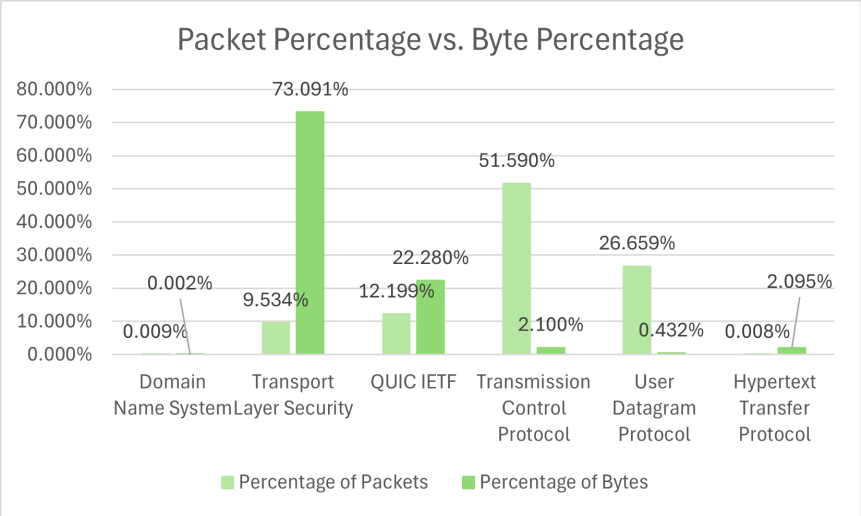

Figure 3: Visualization of protocol usage across student-accessed websites

4 Results

DNS traffic accounted for an extremely small number of overall network activity, with packet and byte percentages close to 0%. Give that DNS is responsible for converting IP addresses into human-readable addresses, it will show up primarily while looking up sites. While students frequently look up sites, it does not account for a large portion of activity on a network. Most activity comes from interaction between devices and the site following look-up.

TLS traffic accounted for only 9.534% of all packets, but 73.091% of all bytes. The variation in these two percentages can be explained with an understanding of what TLS does. TLS is responsible primarily for ensuring encrypted communication between devices. The amount of data encrypted through TLS is mass, given the amount of encrypted sites that students access over a secure network such as campus Wi-Fi. The maximum number of bytes that can be sent through TLS is called the "maximum fragment length," and is set at 2^{14} bytes by default [9]. This indicates that in the data collected, applications were sending data in large encrypted chunks, rather than in smaller packets. However, the byte percentage is massive because most sites were encrypted, and secure.

QUIC IETF traffic accounted for 12.199% of packets and 22.280% of bytes. Generally, QUIC sends packets of approximately 1200 bytes [10],

useful for quick response times and low latency. In this study, QUIC was observed in various streaming and real-time communication platforms such as YouTube, Discord, and Shared Google Docs. QUIC sends packets with large byte counts, and is meant to maximize these large numbers of data for its supported services. QUIC does not have a large number of control packets, which is why it seems to be less frequent than other protocols. However, its role in student usage is vital to ensuring real-time quick and reliable communication.

TCP traffic accounts for 51.590% of packets but only 2.1% of bytes. This can be explained with an understanding of what TCP does. TCP is a control protocol, meaning hundreds of control packets are sent between the device and server. For instance, Figure 4 depicts the three-way handshake that occurs when establishing a connection over a network [11]. The beginning of this process involves the client device sending a synchronize (SYN) packet to the server is trying to connect to which requests the connection. Then, the server sends back a synchronize-acknowledge (SYN/ACK) packet over to the client device, acknowledging and responding to said connection request. Finally, the client device sends over an acknowledgment packet (ACK), which acknowledges and establishes the connection. This process alone contains a small number of bytes, but requires 3 packets to perform. This is only one way in which TCP carries out its duties as a control protocol. There are various other control packets sent between the client and server that TCP takes care of, all of which contain a small number of bytes.

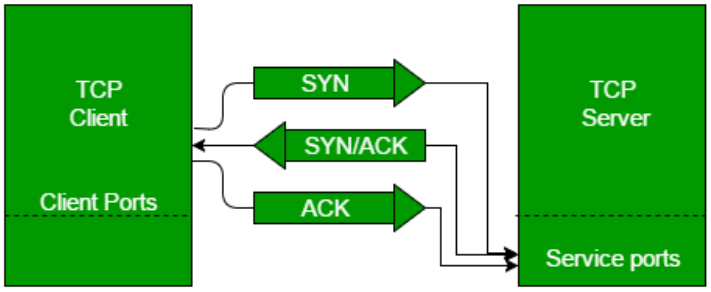

Figure 4: Illustration of the TCP 3-way handshake process [11].

UDP traffic accounts for 26.659% of packets, and only 0.432% of bytes. Once again, these results can be explained once an understanding of what UDP does is explained in the context of this study. UDP is primarily useful

when streaming in real-time. Of the applications and servers analyzed, this includes online video game streaming, Zoom calls, and Microsoft Teams calls. UDP is useful for quick response times, which is useful for these real-time streaming applications. UDP is known for sending packets very quickly between a client and a server, which explains the large percentage of packets. These packets contain a small number of bytes, and are constantly sent back and forth. This explains the small byte percentage and large packet percentage.

HTTP is the final protocol analyzed, and it accounts for only 0.008% of packets and 2.095% of bytes. Traditional HTTP (specifically HTTP/1.1 over plain TCP) is considered less secure today than its alternatives, given that it transmits data over as plain text as opposed to encrypting it. Many services no longer use it due to its lack of security, and have switched to HTTPS, which layers TLS encryption over HTTP to ensure confidentiality when connecting to a network. The lack of HTTP indicates a secure network, and security within the sites students visit.

These percentages can give us a better understanding of how students use TCU's network and which protocols are associated with the websites they visit. The most common network network protocols used can vary depending on whether the analysis focuses on the number of packets or the number of bytes. Overall, it can be observed that protocols such as TLS and TCP appear more frequently, ensuring a secure and reliable network, which directly answers the first and third research questions. It can be inferred that due to the high number of encrypted traffic that Wireshark picked up, most services used by students are encrypted over the network.

This study found that different types of network activities can product various traffic patterns and bandwidth demands. Protocols like TLS and QUIC, were used mainly by modern web and streaming services, accounted a large percentage of total bytes, with moderate packet counts. This indicates that these services transfer large amounts of encrypted or compressed data efficiently in a smaller number of packets. On the other hand, TCP and UDP displayed opposite trends: TCP transferred the most packets due to its role in connection management and control (e.g., three-way handshakes). However, these packets carried small bytes of data. UDP is often used in real-time services like Zoom and online gaming, and it showed a high packet frequency but an extremely low byte volume. This is due to the rapid exchange of lightweight packets for quicker speeds. DNS and HTTP were barely present, with DNS used only in initial lookups and HTTP being replaced by encrypted protocols. These findings indicate that real-time and streaming applications shape traffic by prioritizing speed and reliability,

while academic and general browsing depends more on secure, high-volume encrypted transfer of data, which answers the third question of this study.

5 Conclusions

As the campus network at TCU expands, so does the number of users. This results in increased network activities of all types that needs to be consistently monitored to uphold its security. While the campus network cannot be ethically analyzed, typical everyday student network usage can be simulated through accessing typical academic websites and applications on a device. This data and finding could be key to providing more insight into how secure and efficient are typical websites and applications used commonly as students. The proposed method of analyzing this traffic used Wireshark to capture packets from each website and application. Six key protocols including QUIC/IETF, TLS, DNS, TCP, UDP and HTTP were found across packets while accessing each website and application. Among these protocols, TCP and UDP were the most common protocols among websites and applications tested. Furthermore, TLS had the highest percent bytes which reflects the fact that most websites and applications use encryption for better security. QUIC/IETF had the second highest percent bytes, which indicates that most of the websites and applications tested use QUIC for fast and reliable connections. Overall, from the research conducted, it can be concluded that TLS and QUIC/IETF had the highest percent bytes, which suggests that many common TCU student network activities are secure and more efficient when it comes to bandwidth usage.

With more research in the future, the number of websites and applications in this study can be expanded. Futhermore, in the future it would also be beneficial to conduct a survey among TCU students to update the websites and applications analyzed. With these changes being implemented, they will improve the accurate reflection of student network activity.

References

[1] K.-S. Shim, I. Sohn, E. Lee, W. Seok, and W. Lee, "Enhance the ICS Network Security Using the Whitelist-based Network Monitoring Through Protocol Analysis", JWE, vol. 20, no. 1, pp. 1–32, Feb. 2021.

[2] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco, 1979.

[3] *DNS-BC: Fast, Reliable and Secure Domain Name System Caching System Based on a Consortium Blockchain*, Sensors (Basel, Switzerland), vol. 23, no. 14, pp. 6366-, 2023, doi: 10.3390/s23146366.

[4] S. Turner, "Transport Layer Security," *IEEE Internet Computing*, vol. 18, no. 6, pp. 60–63, Nov.–Dec. 2014, doi: 10.1109/MIC.2014.126.

[5] M. Kosek, T. Shreedhar, and V. Bajpai, "Beyond QUIC v1: A first look at recent transport layer IETF standardization efforts," *IEEE Communications Magazine*, vol. 59, no. 4, pp. 24–29, Apr. 2021, doi: 10.1109/MCOM.001.2000877.

[6] J. Postel, "Transmission Control Protocol," RFC 793, Internet Engineering Task Force (IETF), Sept. 1981. [Online]. Available: `https://datatracker.ietf.org/doc/html/rfc793`

[7] J. Postel, "User Datagram Protocol," RFC 768, Internet Engineering Task Force (IETF), Aug. 1980. [Online]. Available: `https://datatracker.ietf.org/doc/html/rfc768`

[8] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, RFC Editor, United States, June 1999, doi: 10.17487/RFC2616.

[9] "Record Sizes," *s2n-tls Usage Guide*, Amazon Web Services, [Online]. Available: `https://aws.github.io/s2n-tls/usage-guide/ch08-record-sizes.html`. Accessed: Apr. 30, 2025.

[10] L. Oltmans, "The Illustrated QUIC Connection," *xargs.org*, [Online]. Available: `https://quic.xargs.org/#open-all`. Accessed: Apr. 30, 2025.

[11] A. Prasad, "TCP 3-Way Handshake Process," *GeeksforGeeks*, Dec. 27, 2024. [Online]. Available: `https://www.geeksforgeeks.org/tcp-3-way-handshake-process/`

Appendix A: Protocol Aggregation and Percentages Calculations in Excel

To calculate the total number of packets and total number of bytes for each network protocol across all captures, Microsoft Excel was used. Each website and application visited had its own spreadsheet tab that listed the protocol breakdown copied as a .csv from Wireshark's protocol hierarchy under the statistics tab.

The 'SUMIF' function in Excel was used to add all data for matching protocol names across all sheets. For example, to add the total number of packets for the protocol in cell 'A2' from the "YouTube" sheet with the pasted Youtube Protocol Hierarchy capture, the following formula was used:

```
=SUMIF('YouTube'!A:A, A2, 'YouTube'!C:C)
```

Additionally, to calculate the total number of bytes for the same protocol from the same sheet:

```
=SUMIF('YouTube'!A:A, A2, 'YouTube'!E:E)
```

Once the total number of packets and bytes were calculated for each protocol, percent of packets and percent of bytes were calculated next to understand their frequency. These were calculated using the following formula:

```
=B2/SUM(B:B)   // Percentage of total packets in column B
=D2/SUM(D:D)   // Percentage of total bytes in column D
```

This process was repeated across all protocols analyzed.