CAB222 Networks
Project Report

Student and Group Details			
Group number	**29**		
Student name		**Student number**	**Contribution**
Tanvi Sharma		**11758970**	**TCP Protocol – Normal Behaviour**
Harley Bishop			
Mehak Aggarwal		**11892323**	

Only one (1) student should submit this report on behalf of the group. To help your marker enter the grades, please provide the details of the group member who submitted the assignment.			
Student name		**Student number**	

Instructions

You must use this template to complete your Group Project Report. Complete all the fields and then submit on Canvas using the instructions provided in both the assignment specification, under "Assessment 2" on Canvas. The fields are not a fixed size for each question, so you can increase or decrease their size, or expand them depending on your needs. Do NOT modify the margins, font and spacing of this document, i.e.

- **Font: Arial, size 10**
- **Spacing:1.15**
- **Margins: top 3.65cm, bottom 2cm, sides 2 cm**

Do NOT delete any of the sections of the template.

This assignment is completed in groups of three (3) students. Your report must be three to four (3 – 4) pages long. That is no less than three (3) and no more than four (4) pages. Please note the page limit excludes this cover page and any references.

Note:

- **Submission must be in PDF format.**
- **Ensure you insert your group number on the cover page.**

1. TCP Protocol – Normal Behaviour

Provide a detailed and comprehensive explanation of the normal operation of the TCP protocol.

The Transmission Control Protocol (TCP) is a connection-oriented protocol that ensures reliable data transmission between two endpoints over an IP network. It provides mechanisms for establishing a connection, ensuring the integrity and order of transmitted data, and terminating the connection gracefully.

TCP Handshake Process
TCP uses a three-way handshake to establish a connection between the client and the server:

1. **SYN (synchronize)**: The client initiates the connection by sending a SYN packet to the server. This packet includes an initial sequence number.
2. **SYN-ACK (synchronize-acknowledge)**: The server responds with a SYN-ACK packet, which acknowledges the client's SYN and includes the server's own sequence number.
3. **ACK (acknowledge)**: The client sends an ACK packet back to the server, acknowledging the server's SYN-ACK. At this point, the connection is established, and both the client and server are ready to start exchanging data.

This handshake process ensures that both sides synchronize their sequence numbers and are prepared to send and receive data.

Data Transmission
Once the connection is established, TCP ensures reliable data transmission using sequence and acknowledgment numbers:

- **Sequence Numbers**: Each byte of data is assigned a sequence number, allowing the receiver to reassemble the data in the correct order.
- **Acknowledgment Numbers**: The receiver uses acknowledgment numbers to confirm which bytes have been received successfully. For example, if the receiver sends an ACK for sequence number 501, it means that all data up to byte 500 has been successfully received.
- **Window Size (Flow Control)**: TCP uses a sliding window mechanism to control the amount of data that can be sent before needing an acknowledgment. The receiver advertises a window size that indicates how much data it can receive without being overwhelmed.

Connection Termination

TCP uses a **four-way handshake** to terminate a connection:

1. The client sends a **FIN** packet to indicate that it has finished sending data.
2. The server responds with an **ACK** to acknowledge the FIN and then sends its own **FIN** when it is ready to close the connection.
3. The client sends a final **ACK** to confirm the server's FIN, completing the termination process.

This graceful termination ensures that both sides are aware that the connection is closed and that no further data will be exchanged.

Congestion Control and Flow Control

TCP employs several mechanisms to ensure that the network is not overwhelmed by excessive traffic:

- **Congestion Control**: TCP uses algorithms like **slow start**, **congestion avoidance**, **fast retransmit**, and **fast recovery** to adjust the rate of data transmission based on network conditions.
- **Flow Control**: TCP ensures that the sender does not overwhelm the receiver by sending more data than the receiver can handle. This is achieved through the advertised window size, which is dynamically adjusted based on the receiver's buffer capacity.

Conclusion

The TCP protocol is designed to provide reliable, ordered, and error-checked delivery of a stream of bytes between applications running on hosts in a network. Its connection-oriented nature, combined with mechanisms like error detection, retransmission, flow control, and congestion control, ensures that data is delivered accurately and efficiently across unreliable networks.

2. Capture File – Anomalies
2.1 Provide a detailed and specific explanation of all anomalies identified within the provided capture file.
2.2 Which TCP header fields are impacted? Identify the impacted fields. Provide specific packets from the capture file that display the anomaly.
2.3 Are there any anomalies or interesting observations in the other layers? Identify the impacted fields and protocols if applicable. Provide specific packets from the capture file that display the anomaly if applicable.
2.4 Based on the evidence and anomalies discussed above, identify, and explain the type of attack that has likely occurred.

3. Mitigation Strategy 1 (on-premises)
3.1 Based on your own research, identify an appropriate mitigation strategy. This must be technical in nature. This solution must be 'on-premises' (i.e., not a vendor managed cloud-based solution)
3.2 For the identified mitigation strategy, provide a detailed technical explanation of the proposed strategy and where it operates in relation to the layers of the OSI model.
3.3 For the identified mitigation strategy, explain its associated benefits and limitations.

4. Mitigation Strategy 2 (outsourced to a third-party vendor)
4.1 Based on your own research, identify an appropriate mitigation strategy. This solution must be outsourced to a vendor. Identify and explain the solution that you require the vendor to provide.
4.2 You will require a Service Level Agreement (SLA) to be in place with the vendor. Justify the need for implementing the SLA. Discuss the relevant provisions that should be included in the SLA. Identify and justify any relevant Key Performance Indicators (KPIs) that should be included in the SLA.

5. References (excluded from page limit)

5.1 List any references in this section.

Pahdye, J., & Floyd, S. (2001). On inferring TCP behavior. *ACM SIGCOMM Computer Communication Review*, 31(4), 287-298. https://doi.org/10.1145/964723.383083

Hsu, F., Hwang, Y., Tsai, C., Cai, W., Lee, C., & Chang, K. (2016). TRAP: A three-way handshake server for TCP connection establishment. *Applied Sciences*, 6(11), 358. https://doi.org/10.3390/app6110358

Nagle, J. (1984). Congestion control in IP/TCP internetworks. *ACM SIGCOMM Computer Communication Review*, 14(4), 11-17. https://doi.org/10.1145/1024908.1024910

Samaraweera, N. K. G., & Fairhurst, G. (1998). Reinforcement of TCP error recovery for wireless communication. *ACM SIGCOMM Computer Communication Review*, 28(2), 30-38. https://doi.org/10.1145/279345.279348

Han, B., & Billington, J. (2002). Validating TCP connection management. *Proceedings of the Conference on Application and Theory of Petri Nets: Formal Methods in Software Engineering and Defense Systems*, 47-55. https://doi.org/10.5555/846335.846341

Rojviboonchai, K., & Aida, H. (2004). An evaluation of multi-path transmission control protocol (M/TCP) with robust acknowledgement schemes. *IEICE Transactions on Communications*, E87-B(9), 2699-2707. https://search.ieice.org/bin/summary.php?id=e87-b_9_2699

Balakrishnan, H., Padmanabhan, V. N., Seshan, S., & Katz, R. H. (1997). A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6), 756-769. https://doi.org/10.1109/35.650137

Meyer, D., & Zobrist, G. (1990). TCP/IP versus OSI. *IEEE Potentials*, 9(1), 16-19. https://doi.org/10.1109/45.46812