

Computer Communications and Networks (COMN)

2019/20, Semester 2

Assignment Part 1 Results Sheet

| | |
|-----------------------|-----------|
| Forename and Surname: | Joao Maio |
| Matriculation Number: | s1621503 |

Question 1 – Number of retransmissions and throughput with different retransmission timeout values with stop-and-wait protocol. For each value of retransmission timeout, run the experiments for **5 times** and write down **average number of retransmissions** and **average throughput**.

| Retransmission timeout (ms) | Average number of re-transmissions | Average throughput (Kilobytes per second) |
|-----------------------------|------------------------------------|---|
| 5 | 1112.6 | 46.27 |
| 10 | 705.2 | 43.96 |
| 15 | 381.2 | 42.59 |
| 20 | 215.8 | 40.03 |
| 25 | 209.6 | 37.97 |
| 30 | 198.2 | 36.32 |
| 40 | 207.0 | 33.51 |
| 50 | 193.4 | 31.37 |
| 75 | 194.4 | 26.80 |
| 100 | 192.4 | 23.47 |

Question 2 – Discuss the impact of retransmission timeout value on number of retransmissions and throughput. Indicate the optimal timeout value from communication efficiency viewpoint (i.e., the timeout that minimizes the number of retransmissions and keeps the throughput as high as possible).

Retransmission time is inversely proportional to the average number of retransmissions **and** average throughput. While a lower retransmission value may increase the throughput, it increases network traffic unnecessarily. In this case, the round-trip time was 20ms, meaning that there may be up to 4 unnecessary packet retransmissions if the retransmission time is set at 5ms – highly inefficient.

Choosing a retransmission time of **20ms** seems to be a good balance between number of re-transmissions and throughput, since increasing the retransmission time further does not lower the average number of re-transmissions by much, yet greatly reduces throughput. Knowing in advance that the network round-trip time was 20ms, this value makes sense, since it would allow just enough time for packets or responses to be considered “lost” - assuming a perfect network environment and low computational overhead – before re-sending that same packet regardless of whether it was lost or the corresponding ACK packet was lost.