**EE4204 Computer Networks**

**Socket Programming Assignment Report**

Jin Shuyuan A0162475B

1. **Implementation**

My code implements UDP-based client-server socket program for transferring a large file in small data units. I implement ARQ with stop-and-wait and chose to handle retransmission using negative acknowledgement.

The error simulation is done on the server side. I generate a random number between 0 and 1 in server. If the random number is less than the ERROR\_PROB, I pretend that the server cannot receive the packet and sends back a NACK. If the random number is larger or equal to ERROR\_PROB, the server calls recvfrom function and sends back an ACK. On the client side, if it receives NACK, the data unit is retransmitted.

1. **Socket Programming Tools and Functions**

|  |  |
| --- | --- |
| TCP | UDP |
| requires connection establishment | doesn’t have connection establishment and there is no connection state |
|  |  |

1. **Performance Plots**

Assumptions:

The file length is 29896 bytes. We assume that data transfer between different computers is similar to the data transfer on the same machine using localhost. And to simulate lost or damaged packet, I generate a random number on the server side to decide whether the packet transfer has error. If there is a simulated error, I let the server pretend that it didn’t receive the packet and send back a NACK.

1. Transfer time vs error probability (0 and other values of your choice)

|  |  |  |
| --- | --- | --- |
| error probability | transfer time(ms) | data unit size(bytes) |
| 0 | 6.17 | 500 |
| 0.1 | 6.376 | 500 |
| 0.2 | 6.421 | 500 |
| 0.3 | 6.477 | 500 |
| 0.4 | 6.585 | 500 |
| 0.5 | 6.868 | 500 |
| 0.6 | 7.121 | 500 |
| 0.7 | 7.678 | 500 |
| 0.8 | 9.544 | 500 |

Fig 1

1. Throughput vs error probability (0 and other values of your choice)

|  |  |  |
| --- | --- | --- |
| error probability | throughput(Mbps) | data unit size (bytes) |
| 0 | 38.764 | 500 |
| 0.1 | 37.512 | 500 |
| 0.2 | 37.249 | 500 |
| 0.3 | 36.927 | 500 |
| 0.4 | 36.321 | 500 |
| 0.5 | 34.825 | 500 |
| 0.6 | 33.587 | 500 |
| 0.7 | 31.151 | 500 |
| 0.8 | 25.06 | 500 |

Fig 2

1. Transfer time vs data unit size (error probability = 0, error probability = value of your choice)

|  |  |  |
| --- | --- | --- |
| data unit size(bytes) | transfer time(ms) | error probability |
| 150 | 17.173 | 0 |
| 250 | 9.384 | 0 |
| 350 | 8.515 | 0 |
| 450 | 5.754 | 0 |
| 550 | 4.973 | 0 |
| 650 | 4.581 | 0 |
| 750 | 4.226 | 0 |
| 850 | 3.882 | 0 |
| 950 | 3.375 | 0 |

Fig 3

|  |  |  |
| --- | --- | --- |
| data unit size(bytes) | transfer time(ms) | error probability |
| 150 | 20.753 | 0.5 |
| 250 | 10.944 | 0.5 |
| 350 | 8.925 | 0.5 |
| 450 | 5.735 | 0.5 |
| 550 | 5.384 | 0.5 |
| 650 | 4.947 | 0.5 |
| 750 | 4.325 | 0.5 |
| 850 | 3.972 | 0.5 |
| 950 | 3.785 | 0.5 |

Fig 4

1. Throughput vs data unit size (error probability = 0, error probability = value of your choice)

|  |  |  |
| --- | --- | --- |
| data unit size(bytes) | throughput(Mbps) | error probability |
| 150 | 13.927 | 0 |
| 250 | 25.488 | 0 |
| 350 | 28.089 | 0 |
| 450 | 41.567 | 0 |
| 550 | 48.095 | 0 |
| 650 | 52.21 | 0 |
| 750 | 56.596 | 0 |
| 850 | 61.612 | 0 |
| 950 | 70.867 | 0 |

Fig 5

|  |  |  |
| --- | --- | --- |
| data unit size(bytes) | throughput(Mbps) | error probability |
| 150 | 11.525 | 0.5 |
| 250 | 21.855 | 0.5 |
| 350 | 26.798 | 0.5 |
| 450 | 41.705 | 0.5 |
| 550 | 44.423 | 0.5 |
| 650 | 48.348 | 0.5 |
| 750 | 55.301 | 0.5 |
| 850 | 60.216 | 0.5 |
| 950 | 63.19 | 0.5 |

Fig 6

1. **Descriptions**

Based on transfer time vs error probability plot (Fig 1), we can see that the larger the error probability, larger the transfer time will be. Theoretically, for stop-and-wait, U=(1-errorProb)/1+2a. And the total transfer time would be k/U, where k is a constant number given fixed file size. From the formula, we can conclude that transfer time is multiplicative inverse for error probability. From Fig 1, transfer time is roughly inversely proportional to the error probability. And this complies with the theoretical analysis.

The second experiment is throughput vs error probability. Throughput = total size/transfer time = (total size \* U)/k. So, the larger the error probability is, the smaller the throughput should be. This is exactly what is shown in Fig 2.

Fig 3 plots transfer time vs data unit size with 0 error probability and Fig 4 plots transfer time vs data unit size with 0.5 error probability. Transfer time decreases with increasing data unit size. Theoretically, U = P/(1+2a), a = Tp/Tf and total transfer time is k/U. So transfer time = m\*Tp/Tf + b, where m and b are constants given fixed error probability and file size. When data unit increases, Tf increases, transfer time decreases. Thus, the plotted curve is consistent with theory.

Fig 5 and Fig 6 plots throughput vs data unit size given 0 and 0.5 error probability. Throughput = total size/transfer time. Similar to analysis of Fig 2, here, throughput should increase with larger data units.