CIS549 – Project 5

DeSanctis/Finn/Jarčev/Schwyzer

Problem 1:

From the throughput chart, we can see that each UE obtains roughly equal throughput with total throughput being equal to the sum of all 3 UEs.

A screenshot of a social media post

Description automatically generated

From the download stats file, we can validate some of the entries:

time celId IMSI frame sfrm RNTI mcsTb size1 mcs2 sz2 Nprb

1.551 1 3 156 2 1 5 1095 0 0 98

1.624 1 2 163 5 3 25 7167 0 0 98

1.625 1 2 163 6 3 20 4904 0 0 98

1.626 1 2 163 7 3 20 4904 0 0 98

1.627 1 1 163 8 2 15 3542 0 0 97

1.628 1 1 163 9 2 15 3542 0 0 97

1.629 1 1 163 10 2 15 3542 0 0 97

1.63 1 1 164 1 2 10 1980 0 0 99

1.631 1 1 164 2 2 10 1980 0 0 99

For instance, we take the frame ID (minus 1 because the subframes start with 1 in the file, but in our program they start with 0) and add to the subframe (minus 1 because the subframes are listed as 1-10, but in our program, they are 0-9). So, Frame 163, subframe 10 is row 1629 in our validation sheet and Frame 164, subframe 1 is row 1630. For IMSI=1 (came from the UE2.txt parameter file), we can see the MCS value change from 15 to 10. Below is a subset of our validation spreadsheet. It shows that we can expect the MCS value to change from 15 to 10.

|  |  |  |  |
| --- | --- | --- | --- |
| frame | UE1(3) MCS | UE2(1) MCS | UE3(2) MCS |
| 1623 | 5 | 15 | 25 |
| 1624 | 1 | 10 | 20 |
| 1625 | 5 | 15 | 25 |
| 1626 | 1 | 10 | 20 |
| 1627 | 5 | 15 | 25 |
| 1628 | 1 | 10 | 20 |
| 1629 | 5 | 15 | 25 |
| 1630 | 1 | 10 | 20 |
| 1631 | 5 | 15 | 25 |
| 1632 | 1 | 10 | 20 |
| 1633 | 5 | 15 | 25 |

Let’s take another example for IMSI=3 (came from the UE1.txt parameter file):

time celId IMSI frame sfrm RNTI mcsTb size1 mcs2 sz2 Nprb

1.042 1 3 105 3 1 5 349 0 0 32

1.042 1 1 105 3 2 10 645 0 0 32

1.042 1 2 105 3 3 20 1572 0 0 32

1.139 1 1 114 10 2 10 645 0 0 32

1.139 1 2 114 10 3 20 1572 0 0 32

1.139 1 3 114 10 1 1 145 0 0 32

1.14 1 3 115 1 1 1 217 0 0 47

1.14 1 1 115 1 2 10 935 0 0 47

1.141 1 3 115 2 1 1 453 0 0 98

1.142 1 3 115 3 1 1 453 0 0 98

1.143 1 3 115 4 1 5 1095 0 0 98

1.236 1 2 124 7 3 20 4904 0 0 98

1.237 1 1 124 8 2 10 1980 0 0 99

1.238 1 1 124 9 2 10 1980 0 0 99

Here you can see Frame 115, subframe 3 (row 1142 in the validation sheet) and Frame 115, subframe 4 (row 1143). The MCS value changes from 5 to 1, which is what we see in the validation sheet:

|  |  |  |  |
| --- | --- | --- | --- |
| frame | UE1(3) MCS | UE2(1) MCS | UE3(2) MCS |
| 1138 | 1 | 10 | 20 |
| 1139 | 5 | 15 | 25 |
| 1140 | 1 | 10 | 20 |
| 1141 | 5 | 15 | 25 |
| 1142 | 1 | 10 | 20 |
| 1143 | 5 | 15 | 25 |
| 1144 | 1 | 10 | 20 |
| 1145 | 5 | 15 | 25 |
| 1146 | 1 | 10 | 20 |
| 1147 | 5 | 15 | 25 |
| 1148 | 1 | 10 | 20 |

Final example for IMSI=2 (came from the UE3.txt parameter file):

time celId IMSI frame sfrm RNTI mcsTb size1 mcs2 sz2 Nprb

3.963 1 2 397 4 3 20 2385 0 0 48

3.964 1 2 397 5 3 20 2385 0 0 48

3.964 1 3 397 5 1 5 533 0 0 48

3.965 1 2 397 6 3 20 2385 0 0 48

3.965 1 3 397 6 1 1 217 0 0 47

3.966 1 3 397 7 1 1 217 0 0 47

3.966 1 2 397 7 3 20 2385 0 0 48

3.967 1 2 397 8 3 25 3422 0 0 47

3.967 1 3 397 8 1 1 217 0 0 47

3.968 1 2 397 9 3 25 3422 0 0 47

Here you can see Frame 397, subframe 7 (row 3966 in the validation sheet) and Frame 397, subframe 8 (row 3967). The MCS value changes from 20 to 25, which is what we see in the validation sheet:

|  |  |  |  |
| --- | --- | --- | --- |
| frame | UE1(3) MCS | UE2(1) MCS | UE3(2) MCS |
| 3963 | 5 | 15 | 25 |
| 3964 | 1 | 10 | 20 |
| 3965 | 5 | 15 | 25 |
| 3966 | 1 | 10 | 20 |
| 3967 | 5 | 15 | 25 |
| 3968 | 1 | 10 | 20 |
| 3969 | 5 | 15 | 25 |
| 3970 | 1 | 10 | 20 |

Problem 2

Flowchart:

A close up of a logo

Description automatically generated

Validation scenario used to generate Gnuplots:

./waf --run "scratch/prj5\_px --OutputFileName=output/prj --Scenario=3 --NumberUE=1 --Transport=1 --wifiMcs=HtMcs7 --tcpRcvBufBytes=1024000 --DataSizeforTCP=5000000 --DataRateforUDP=200Mb/s --delayValueforRHtoR=5 --delayValueforLte=50 --delayValueforWifi=10 --simTime=2 --aggPath=lteAndWifi --inOrderTimeout=100"

Scenario includes a dropped packet #100

Gnuplots:

We can see the delay is not greater than our timeout period of 100ms.

A screenshot of a social media post

Description automatically generated

We can see some of the packets are out of sequence when they are received by RecvQueue.

A picture containing screenshot

Description automatically generated

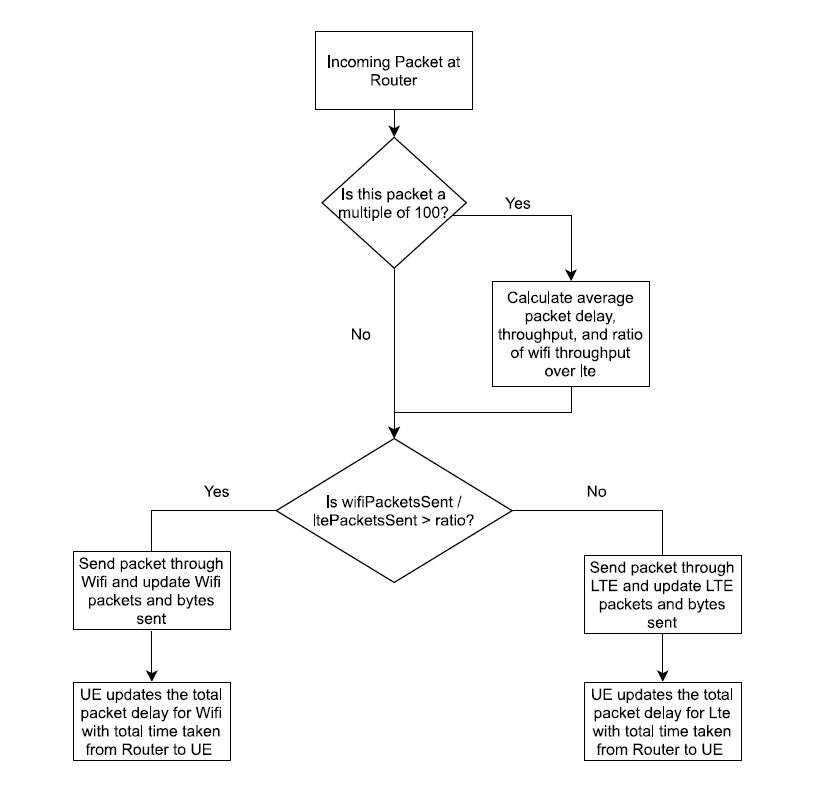
We can see the packets are all in order when sent to the upper layer.

A screenshot of a cell phone

Description automatically generated

**Problem 3:**

Flowchart:



Implementation:

1. Created global variables in lines 73-80:
   1. lteBytesSent
   2. ltePacketsSent
   3. wifiBytesSent
   4. wifiPacketsSent
   5. packetDelayWifi
   6. packetDelayLte
2. Inserted flowchart logic into rtVirtualSend function from lines 607-717
3. Updated the LTE & Wifi Tunnel receive functions to have the UE’s update packetDelayWifi & packetDelayLte with the packet delay for that individual packet.

Validation:

Updated the rtVirtualSend Function to show “LTE packet” or “wifiPacket” in the aggregation scenario to show on the cmd line whether a packet was being sent via lte or via wifi. Then the commands below were used to test the verification that the algorithm was working as intended. The following is the results of the aggregation scenario’s throughput vs. the Wifi & LTE only for 1UE:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RTT(ms) | RWND (Bytes) | MCS | 1 UE | | | | | |
| Per UE | | | | | |
| Wi-Fi TCP | | LTE TCP | | Aggregation TCP | |
| Peak (Mbps) | Avg (Mbps) | Peak (Mbps) | Avg (Mbps) | Peak (Mbps) | Avg (Mbps) |
| 30 ms | 64000 | HtMcs1 | 14.78 | 11.59 |  |  | 14.78 | 10.25 |
| 200 ms | 64000 | HtMcs1 | 5.04 | 2.37 |  |  | 5.04 | 3.71 |
| 30 ms | 1024000 | HtMcs1 | 24.08 | 21.62 |  |  | 81.9 | 14.03 |
| 200 ms | 1024000 | HtMcs1 | 23.86 | 13.56 |  |  | 73.93 | 21.62 |
| 30 ms | 64000 | HtMcs7 | 16.91 | 14.81 | 14.11 | 10.13 | 14.78 | 10.38 |
| 200 ms | 64000 | HtMcs7 | 5.04 | 2.67 | 4.93 | 2.20 | 5.04 | 3.50 |
| 30 ms | 1024000 | HtMcs7 | 114.69 | 80.00 | 72.58 | 50.00 | 158.70 | 66.66 |
| 200 ms | 1024000 | HtMcs7 | 60.03 | 17.78 | 72.46 | 16.67 | 75.60 | 22.22 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| RTT | MCS | 1 UE | | | | | |
| System Level | | | | | |
| Wi-Fi UDP | | LTE UDP | | Aggregation UDP | |
| Peak (Mbps) | Avg (Mbps) | Peak (Mbps) | Avg (Mbps) | Peak (Mbps) | Avg (Mbps) |
| 30 ms | HtMcs1 | 25.20 | 25.09 |  |  | 50.4 | 50.02 |
| 200 ms | HtMcs1 | 25.09 | 25.09 |  |  | 66.19 | 49.97 |
| 30 ms | HtMcs7 | 121.97 | 121.73 | 73.70 | 73.70 | 151.76 | 150.16 |
| 200 ms | HtMcs7 | 121.97 | 121.73 | 73.70 | 73.70 | 178.86 | 158.74 |

Commands used:

TCP –

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_10m\_tcp\_1ue\_m7\_rw1m\_rtt30 --Scenario=3 --NumberUE=1 --wifiMcs=HtMcs7 --Transport=1 --DataSizeforTCP=10000000 --tcpRcvBufBytes=1024000 --delayValueforRHtoR=5 --delayValueforWifi=10 --delayValueforLte=10 --simTime=20 --aggPath=lteAndWifi"

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_10m\_tcp\_1ue\_m1\_rw1m\_rtt30 --Scenario=3 --NumberUE=1 --wifiMcs=HtMcs1 --Transport=1 --DataSizeforTCP=10000000 --tcpRcvBufBytes=1024000 --delayValueforRHtoR=5 --delayValueforWifi=10 --delayValueforLte=10 --simTime=20 --aggPath=lteAndWifi"

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_10m\_tcp\_1ue\_m7\_rw64k\_rtt30 --Scenario=3 --NumberUE=1 --wifiMcs=HtMcs7 --Transport=1 --DataSizeforTCP=10000000 --tcpRcvBufBytes=64000 --delayValueforRHtoR=5 --delayValueforWifi=10 --delayValueforLte=10 --simTime=20 --aggPath=lteAndWifi"

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_10m\_tcp\_1ue\_m1\_rw64k\_rtt30 --Scenario=3 --NumberUE=1 --wifiMcs=HtMcs1 --Transport=1 --DataSizeforTCP=10000000 --tcpRcvBufBytes=64000 --delayValueforRHtoR=5 --delayValueforWifi=10 --delayValueforLte=10 --simTime=20 --aggPath=lteAndWifi"

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_10m\_tcp\_1ue\_m7\_rw1m\_rtt200 --Scenario=3 --NumberUE=1 --wifiMcs=HtMcs7 --Transport=1 --DataSizeforTCP=10000000 --tcpRcvBufBytes=1024000 --delayValueforRHtoR=10 --delayValueforWifi=90 --delayValueforLte=90 --simTime=20 --aggPath=lteAndWifi"

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_10m\_tcp\_1ue\_m1\_rw1m\_rtt200 --Scenario=3 --NumberUE=1 --wifiMcs=HtMcs1 --Transport=1 --DataSizeforTCP=10000000 --tcpRcvBufBytes=1024000 --delayValueforRHtoR=10 --delayValueforWifi=90 --delayValueforLte=90 --simTime=20 --aggPath=lteAndWifi"

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_10m\_tcp\_1ue\_m7\_rw64k\_rtt200 --Scenario=3 --NumberUE=1 --wifiMcs=HtMcs7 --Transport=1 --DataSizeforTCP=10000000 --tcpRcvBufBytes=64000 --delayValueforRHtoR=10 --delayValueforWifi=90 --delayValueforLte=90 --simTime=20 --aggPath=lteAndWifi"

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_10m\_tcp\_1ue\_m1\_rw64k\_rtt200 --Scenario=3 --NumberUE=1 --wifiMcs=HtMcs1 --Transport=1 --DataSizeforTCP=10000000 --tcpRcvBufBytes=64000 --delayValueforRHtoR=10 --delayValueforWifi=90 --delayValueforLte=90 --simTime=20 --aggPath=lteAndWifi"

UDP –

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_50Mb\_udp\_1ue\_m1\_rtt30 --Scenario=3 --NumberUE=1 --Transport=2 --DataRateforUDP=50Mb/s --wifiMcs=HtMcs1 --delayValueforRHtoR=5 --delayValueforWifi=10 --delayValueforLte=10 --simTime=3 --aggPath=lteAndWifi"

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_150Mb\_udp\_1ue\_m7\_rtt30 --Scenario=3 --NumberUE=1 --Transport=2 --DataRateforUDP=150Mb/s --wifiMcs=HtMcs7 --delayValueforRHtoR=5 --delayValueforWifi=10 --delayValueforLte=10 --simTime=3 --aggPath=lteAndWifi"

./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_50Mb\_udp\_1ue\_m1\_rtt200 --Scenario=3 --NumberUE=1 --Transport=2 --DataRateforUDP=50Mb/s --wifiMcs=HtMcs1 --delayValueforRHtoR=10 --delayValueforWifi=90 --delayValueforLte=90 --simTime=3 --aggPath=lteAndWifi"

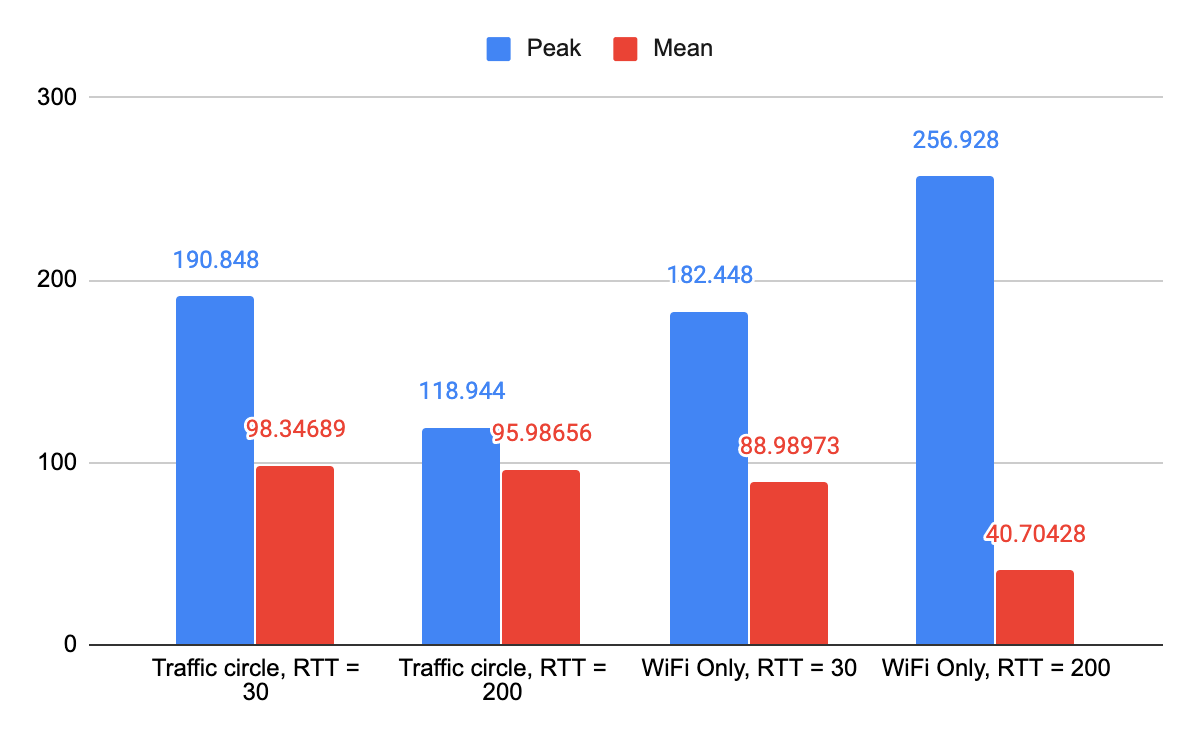
./waf --run "scratch/prj5\_px --OutputFileName=output/prj\_150Mb\_udp\_1ue\_m7\_rtt200 --Scenario=3 --NumberUE=1 --Transport=2 --DataRateforUDP=150Mb/s --wifiMcs=HtMcs7 --delayValueforRHtoR=10 --delayValueforWifi=90 --delayValueforLte=90 --simTime=3 --aggPath=lteAndWifi"

**Problem 4:**

Execute 4 simulations with the input parameters listed below and analyze the average system throughput results.

Instead of splitting traffic to the LTE and Wi-Fi paths, redirect all DL packets toward Wi-Fi path and send all UL packet through LTE path. Compare the TCP average throughput results with the results between the two network path configuration cases (“both DL and UL using Wi-Fi” and “DL using Wi-Fi and UL using LTE”). If there was a performance difference, then what would be the reason?

|  |  |  |
| --- | --- | --- |
| **Connection Type** | **Peak** | **Mean** |
| Traffic circle, RTT = 30 | 190.848 | 98.34689 |
| Traffic circle, RTT = 200 | 118.944 | 95.98656 |
| WiFi Only, RTT = 30 | 182.448 | 88.98973 |
| WiFi Only, RTT = 200 | 256.928 | 40.70428 |



As we can see in the table and graph above, there is some consistency between using the traffic circle of all DL routed via WiFi and UL via LTE and both UL and DL routed via WiFi only at 30ms. In fact, the peaks and averages were very close together, with the WiFi-only routing getting nearly 96% of the peak throughput of the traffic circle, and nearly 91% of the average throughput.

Adding the question of increasing the RTT from 30ms to 200ms, firstly we will start with the traffic circle connection type. In this instance, we saw a significant reduction in peak throughput, which has been consistently shown throughout all ns-3 simulations throughout this course. In this case it is a reduction from almost 191 to 119. Somewhat more surprising is the relatively low reduction in average throughput, from 98 to 96, an almost identical outcome. This is a clearly exhibited advantage of using the two simultaneous paths.

The most surprising result would surely be the WiFi only path with an RTT of 200ms. Here we saw the biggest extremes of throughput behavior. On a granular level, comparing all 15 UEs rather than on the system level, each UE exhibited a curiously high peak throughput of around 80 Mbps. The system peak of over 250Mbps was truly surprising, especially considering the higher RTT. This was a consistently higher number than all other scenarios. And yet, even with the highest of peaks, we also observed the lowest of troughs. WiFi Only with an RTT of 200ms had by far the lowest average throughput; around 45% of the second lowest average throughput (WiFi only at 30ms), and only 41% of the highest (WiFi+LTE at 30ms). While WiFi in general has much lower latency than LTE, this high latency of 200ms (especially for WiFi) may have severely hampered the ability of the connection to perform its UL performance as anticipated.