IIT
Guwahati

CS588:Computer System Lab

(January-May 2023)

Assignment–3:NetworkSimulationusingNS-3

Submitted by:
Kaja Gnana Prakash- 224101027
Tejas Chandra K- 224101052
Ashish Kumar Pal - 224101009

Assignment details:

In this assignment you need to simulate a computer network for a given application using the discrete event network simulator ns-3. You can download the software and documentation of ns-3 from the website https://www.nsnam.org. The assignment will be solved in groups where each group, comprised of 3 members, needs to work on an application assigned to it. The applications' network specifications, the required experiments, and related questions are given in pages 2-7 of this document. Follow the general instructions given below and any specific instructions mentioned in the application description. General Instructions:

1.Each group needs to simulate one application assigned to them and make one single submission on Moodle. Only one member from a group needs to make the submission. The information about the allocation of applications to groups is contained inTable1given below.

2.Install ns-3 on your computer, write programs and simulate the network described in the given application assigned to you, perform the required experiments, and answer the given questions.

3.Use ns-3's Flow Monitor module to collect and store the performance data of network protocols from the simulation. Do not use PCAP + Wireshark for the trace collection. No marks will be awarded if your application uses a PCAP file for trace collection.

4.Submit your set of source code files, along with a report clearly mentioning all observations, explanations, containing the graphs, and all the answers, as a zipped file on Moodle.

5.Write your own source codes and do not copy from any source. Plagiarism and use of unfair means will be penalised by awarding NEGATIVE marks (equal to the maximum marks for the assignment).

Table 1: Allocation of applications to groups

Application No	Group No
1	1,6,11,16,21
2	2,7,12,17,22
3	3,8,13,18
4	4,9,14,19
5	5,10,15,20

Our group number is : 5

So application : 5

Application#5:

Using the network simulator ns-3, study the characteristics of IEEE 802.11. For the purpose of experiment, use the topology as follows. There are 3 nodes in the network located in a straight line at locations 250*i, with i=0,1, 2. Node 0 and Node 2 both have TCP traffic to Node 1 (started randomly within 1 to 5 seconds of starting the simulation). Consider TCP Westwood+ or TCP Hybla for the TCP agents at Node 0 and Node 2, respectively. You Have to run the simulations and measure the following from the trace output (the averages are taken over all the nodes).Do not use the PCAP file for collecting the trace. Use Flow Monitor module in ns-3 for trace collection. No marks will be given if youconsiderPCAPtrace with Wireshark.

1. Average And width spent in transmitting RTS, CTS, and ACK.
2. Average Band width spent in transmittingTCPsegments and TCP acknowledgements.
3. Average Bandwidth Wasted Due To Collisions.
4. TCPthroughput(number of acknowledged bytes per unit time) at each node.

You have to run the simulations for 50 seconds each with different RTS thresholds (i.e. 0, 256, 512 and 1000 bytes) and TCP segment size of 1000 bytes. You can use scripts for trace file analysis and to plot the results.Make appropriate assumptions wherever necessary.

Overview of code:

This is a C++ program that sets up a wireless ad hoc network using the ns-3 network simulator. The program creates three nodes and instals the Internet stack on them. The mobility model is set to a grid position allocator, and the Constant Position Mobility Model is used. The program then creates WiFi devices and sets the MAC and physical layer parameters. TCP applications are installed on nodes 0 and 2, using the OnOffHelper. Finally, a TCP source and sink are set up on nodes 0 and 2 using the PacketSinkHelper. The program also calculates and outputs the throughput of the network every 100 milliseconds using a FlowMonitor. The program is intended to simulate a wireless ad hoc network and measure its performance.

Code Structure and Description:

The program uses various modules provided by the ns-3 library, such as the **mobility module, Wi-Fi module, and flow monitor module.**

The program first sets **default values** for various network parameters such as the **RTS/CTS threshold, fragmentation threshold, and TCP segment size**. It then parses command-line arguments to set the data rate, TCP variant, PHY rate, and other simulation parameters.

The program then sets up the physical layer of the wireless network using the **YansWifiPhyHelper and YansWifiChannelHelper classes**. The mobility module is used to randomly position the nodes in a 2D space.

Next, the **Wi-Fi module is set up with the access point and client nodes.** The program sets up an OnOff application to generate traffic from the client nodes to the access point. A PacketSink application is set up on the access point to receive the traffic.

Finally, the program uses the **flow monitor module to collect statistics** on the data flows in the network. The program calculates the throughput of the network and outputs it to the console. The simulation is run for a fixed period of time, after which the program cleans up and exits.

Outputs:

Output at RTS Threshold set to 0 Bytes:

```
Average bandwidth spent on RTS: 80490 Kbps
Average bandwidth spent on CTS: 79290 Kbps
Average bandwidth spent on ACK: 80490 Kbps
Average bandwidth spent on TCP segments: 36676.9 Kbps
Average bandwidth spent on TCP ACKs: 41848 Kbps
Average bandwidth wasted due to collisions: 1.77275 Kbps
Node 0 TCP throughput: 5.38794 Mbps
Node 1 TCP throughput: 0.288736 Mbps
Node 2 TCP throughput: 6.15161 Mbps

Average throughput: 10.8002 Mbit/s
```

Output at RTS Threshold set to 256 Bytes:

```
Average bandwidth spent on RTS: 84544.1 Kbps
Average bandwidth spent on CTS: 84345.5 Kbps
Average bandwidth spent on ACK: 84544.1 Kbps
Average bandwidth spent on TCP segments: 40733.2 Kbps
Average bandwidth spent on TCP ACKs: 41668.4 Kbps
Average bandwidth wasted due to collisions: 0.399627 Kbps
Node 0 TCP throughput: 5.62106 Mbps
Node 1 TCP throughput: 0.295129 Mbps
Node 2 TCP throughput: 5.73131 Mbps

Average throughput: 10.7619 Mbit/s
```

Output at RTS Threshold set to 512 Bytes:

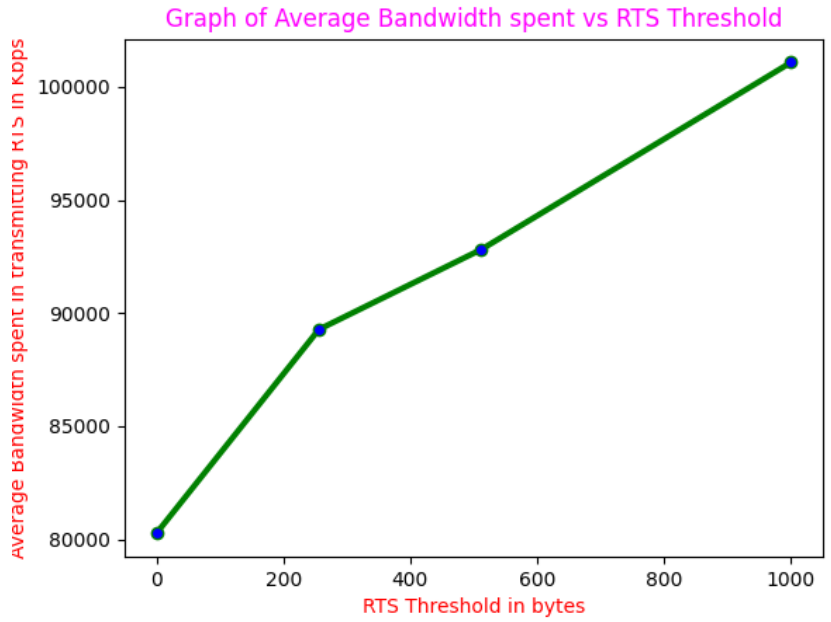
```
Average bandwidth spent on RTS: 89277 Kbps
Average bandwidth spent on CTS: 126374 Kbps
Average bandwidth spent on ACK: 89277 Kbps
Average bandwidth spent on TCP segments: 9099.68 Kbps
Average bandwidth spent on TCP ACKs: 77072.6 Kbps
Average bandwidth wasted due to collisions: 0.939547 Kbps
Node 0 TCP throughput: 1.05791 Mbps
Node 1 TCP throughput: 0.30334 Mbps
Node 2 TCP throughput: 11.0982 Mbps

Average throughput: 11.5235 Mbit/s
```

Output at RTS Threshold set to 1000 Bytes:

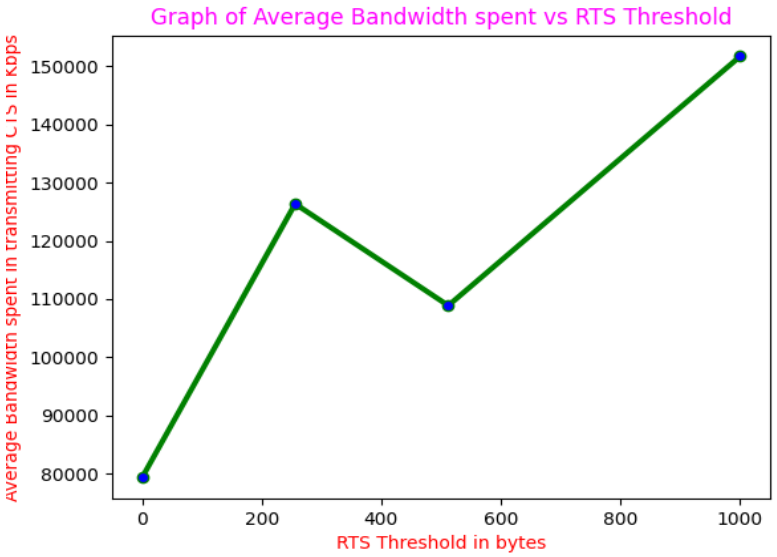
```
Average bandwidth spent on RTS: 99615.2 Kbps
Average bandwidth spent on CTS: 108127 Kbps
Average bandwidth spent on ACK: 99615.2 Kbps
Average bandwidth spent on TCP segments: 1074.23 Kbps
Average bandwidth spent on TCP ACKs: 95820.9 Kbps
Average bandwidth wasted due to collisions: 0.92768 Kbps
Node 0 TCP throughput: 0.111613 Mbps
Node 1 TCP throughput: 0.352244 Mbps
Node 2 TCP throughput: 14.0855 Mbps

Average throughput: 13.467 Mbit/s
```

The following are the plot analysis for each of the bandwidths at each RTS Threshold.

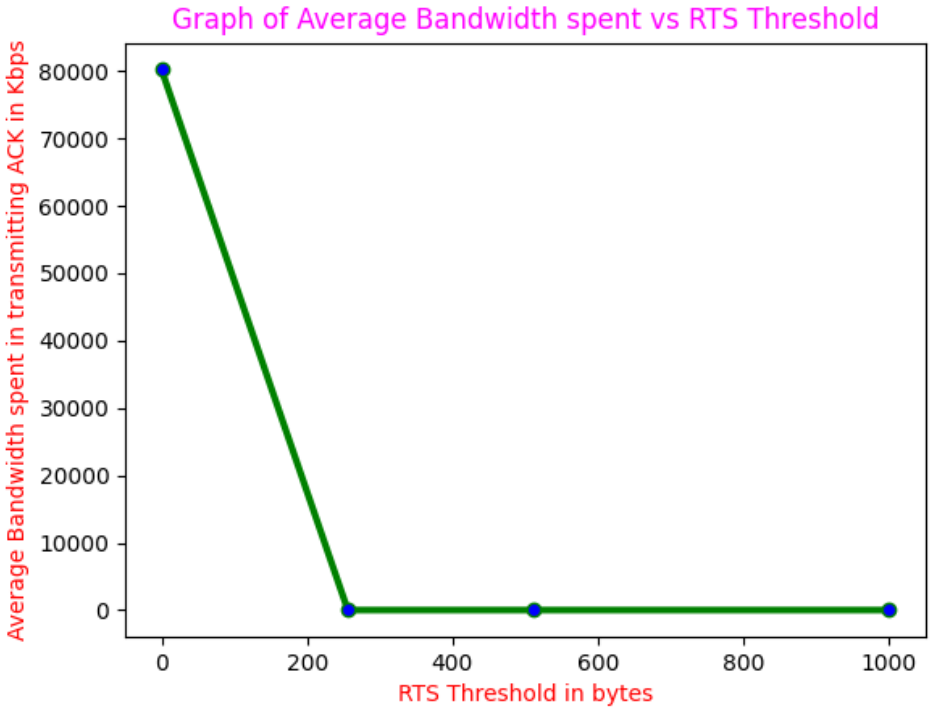
Average Bandwidth Spent on RTS in Kbps vs RTS Thresholds



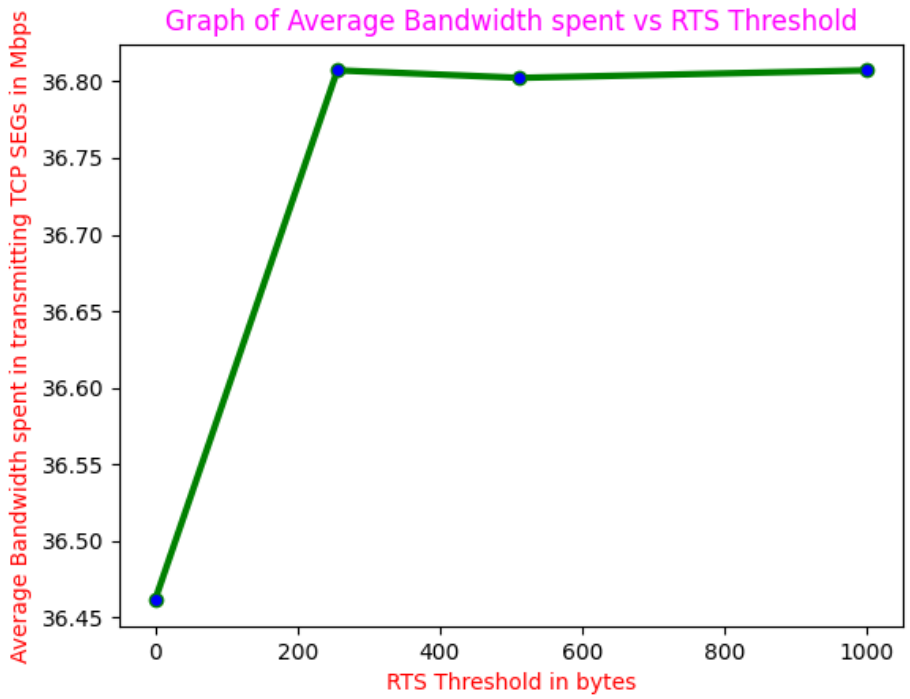
Average Bandwidth Spent on CTS in Kbps vs RTS Thresholds



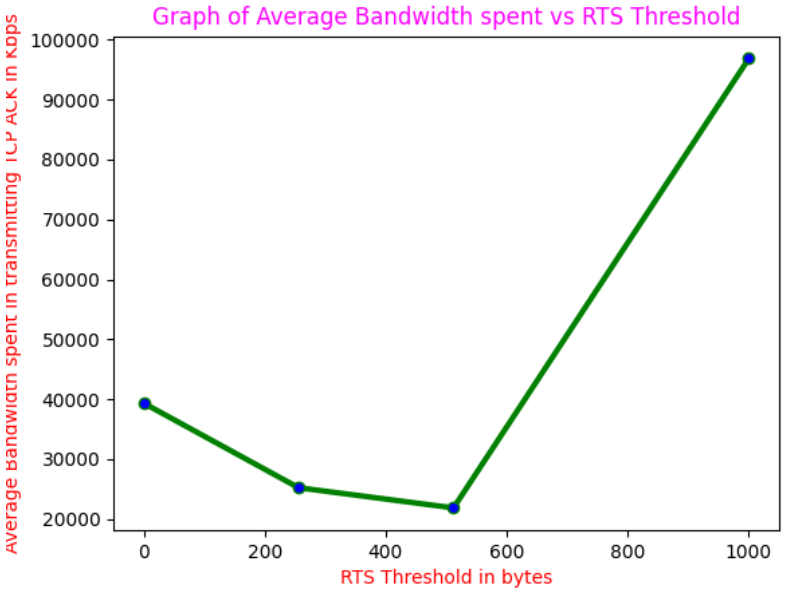
Average Bandwidth Spent on ACK in Kbps vs RTS Thresholds



Average Bandwidth Spent on TCP Segments in Mbps vs RTS Thresholds



Average Bandwidth Spent on TCP ACKs in Kbps vs RTS Thresholds



Average Bandwidth Spent on TCP Throughput in Mbps vs RTS Thresholds