Task 1:-

<u>Experimental Setup:</u>

- The Wireless LAN is operating in Ad-Hoc mode with 6 nodes.
- The nodes move in a 2D random walk in a rectangular area given by (x=-100m, y=-100m) to (x=-100m, y=-100m).
- The channel is the default wireless channel in NS-3
- The physical layer has the default parameters in IEEE 802.11G standard and adaptive rate control is given by AARF algorithm.
- The link layer follows the standard MAC without quality of service control.
- The network layer is standard IPv4 with address range 192.168.1.0/24
- The UDP echo server at node 0 is listening to port 20.
- The UDP echo client at node 5 sends packets at times 2s and 4s.
- The UDP echo client at node 4 sends packets at times 3s and 4s.
- The packet size taken is 1024 bytes
- The reference code is taken from third.cc from ns-3.25\examples\tutorials.

Changes made to the code third.cc:-

- There was no need of p2p or CSMA Nodes so they were removed
- All the nodes were wifi adhoc nodes. They are created by the following command:
  NodeContainer wifiadhocnodes;
  wifiadhocnodes.Create (nWifi);
- The nodes were made adhoc with the following command:
  mac.SetType ("ns3::AdhocWifiMac");

- The echo server at node 0 was listening to port 20 and echo client at node 5 and node 4 sends 2 UDP packets at times 2s, 4s and 3s, 4s. This was done by the following code:

```
ApplicationContainer serverApps = echoServer.Install (wifiadhocnodes.Get (0));
 serverApps.Start (Seconds (1.0));
 serverApps.Stop (Seconds (10.0));

 UdpEchoClientHelper echoClient1 (wifiInterfaces.GetAddress (0), 20);
 echoClient1.SetAttribute ("MaxPackets", UintegerValue (2));
 echoClient1.SetAttribute ("Interval", TimeValue (Seconds (2.0)));
 echoClient1.SetAttribute ("PacketSize", UintegerValue (1024));

 ApplicationContainer clientApps1 =
 echoClient1.Install (wifiadhocnodes.Get (5));
 clientApps1.Start (Seconds (2.0));
 clientApps1.Stop (Seconds (5.0));
```

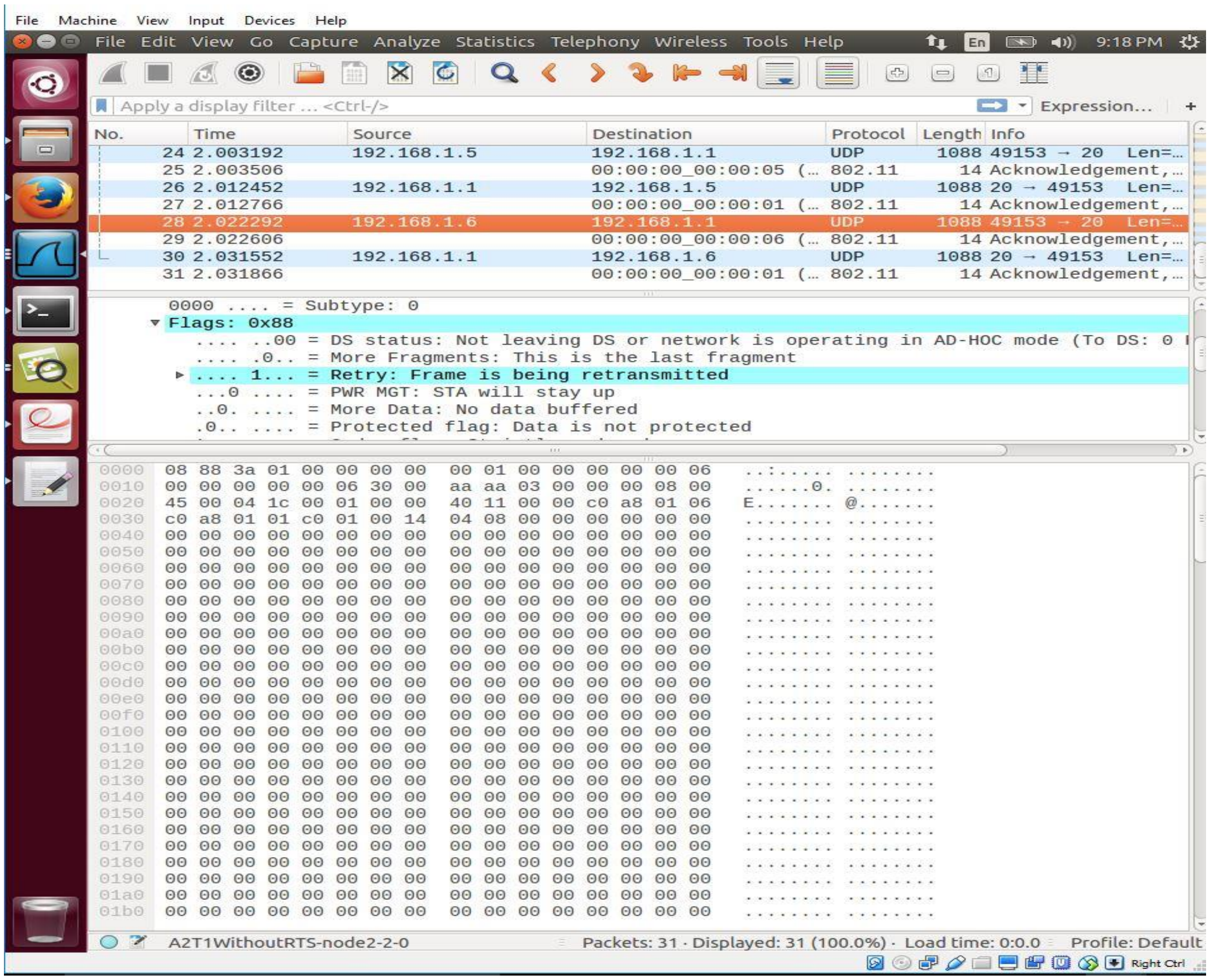
```
UdpEchoClientHelper echoClient2 (wifiInterfaces.GetAddress (0), 20);
echoClient2.SetAttribute ("MaxPackets", UintegerValue (2));
echoClient2.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient2.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps2 =
  echoClient2.Install (wifiadhocnodes.Get (4));
clientApps2.Start (Seconds (3.0));
clientApps2.Stop (Seconds (5.0));
```

- The packet tracer was set up at Node 2 by the following code:

```
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
phy.EnablePcap("A2T1WithoutRTS-node2",adDevices.Get(2),true);
Simulator::Stop (Seconds (10.0));
```

<u>Task 1 Answers:</u>

1. No, all the frames are not acknowledged as there was a collision. As WiFi follows ARQ, the frame that suffered collision will be retransmitted. Apart from the frame which suffered collision all other frames were seen to be acknowledged.
2. Yes, there are collisions in the network. As it can be seen from wireshark trace file that a retransmission flag has been raised which shows the collision. This happens because both node 5 and node 4 transmit at 4s, and they have received their ARP's before this.


File Machine View Input Devices Help
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
En 9:18 PM
Apply a display filter … <Ctrl-/>
Expression…
Right Ctrl

```
No.  Time        Source        Destination              Protocol Length Info
  24 2.003192    192.168.1.5   192.168.1.1              UDP        1088 49153 → 20   Len=…
  25 2.003506                  00:00:00_00:00:05 (…     802.11       14 Acknowledgement,…
  26 2.012452    192.168.1.1   192.168.1.5              UDP        1088 20 → 49153   Len=…
  27 2.012766                  00:00:00_00:00:01 (…     802.11       14 Acknowledgement,…
  28 2.022292    192.168.1.6   192.168.1.1              UDP        1088 49153 → 20   Len=…
  29 2.022606                  00:00:00_00:00:06 (…     802.11       14 Acknowledgement,…
  30 2.031552    192.168.1.1   192.168.1.6              UDP        1088 20 → 49153   Len=…
  31 2.031866                  00:00:00_00:00:01 (…     802.11       14 Acknowledgement,…
```

```
    0000 .... = Subtype: 0
  ▾ Flags: 0x88
      .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 
      .... .0.. = More Fragments: This is the last fragment
    ▸ .... 1... = Retry: Frame is being retransmitted
      ...0 .... = PWR MGT: STA will stay up
      ..0. .... = More Data: No data buffered
      .0.. .... = Protected flag: Data is not protected
```

```
0000  08 88 3a 01 00 00 00 00  00 01 00 00 00 00 00 06   ..:..... ........
0010  00 00 00 00 00 06 30 00  aa aa 03 00 00 00 08 00   ......0. ........
0020  45 00 04 1c 00 01 00 00  40 11 00 00 c0 a8 01 06   E....... @.......
0030  c0 a8 01 01 c0 01 00 14  04 08 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

```
A2T1WithoutRTS-node2-2-0    Packets: 31 · Displayed: 31 (100.0%) · Load time: 0:0.0    Profile: Default
```

<u>*Figure 1: Collision in Ad-hoc mode without RTS/CTS*</u>

3. We can force the nodes to utilize the RTS/CTS procedure by including the following line to the existing code:

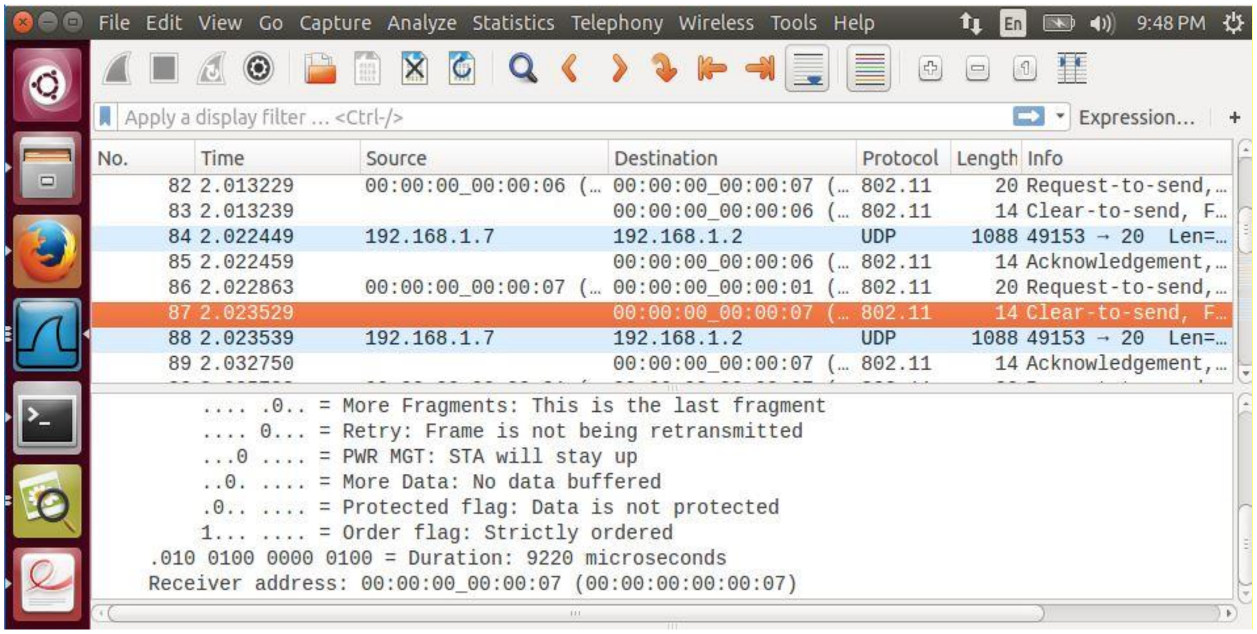
   Config::SetDefault ("ns-3::WifiRemoteStationManager::RtsCtsThreshold", UintegerValue(1));

   The RTS/CTS is enabled by setting the packet size threshold in the program. In our case it is set to 1. So in our case the nodes will be implementing RTS/CTS when the frame size is greater than 1 byte (threshold). In other words it can be said that the RTS/CTS will be enabled all the time in our case. The threshold size can be between 0 to 2347 bytes.

4. a) After the implementation of RTS/CTS from the wireshark trace file it can be seen that there is no collision

b) We implement RTS/CTS to minimize the collisions by making an initial handshake. The RTS frame specifies how much data the node needs to transmit and the CTS frame specifies for how long the channel will be occupied. In case of WiFi Standard CTS is called Network Allocation Vector (NAV).

c) The Network Allocation Vector can be found in the CTS frame as duration. It can be found by going to a trace file and selecting a clear to send frame. After selecting it we can check for the duration field from the drop down to check what the network allocation vector is.



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
9:48 PM
Apply a display filter … <Ctrl-/>
Expression…

```
No.  Time       Source                 Destination            Protocol Length Info
  82 2.013229   00:00:00_00:00:06 (…  00:00:00_00:00:07 (…  802.11     20 Request-to-send,…
  83 2.013239                          00:00:00_00:00:06 (…  802.11     14 Clear-to-send, F…
  84 2.022449   192.168.1.7            192.168.1.2            UDP      1088 49153 → 20  Len=…
  85 2.022459                          00:00:00_00:00:06 (…  802.11     14 Acknowledgement,…
  86 2.022863   00:00:00_00:00:07 (…  00:00:00_00:00:01 (…  802.11     20 Request-to-send,…
  87 2.023529                          00:00:00_00:00:07 (…  802.11     14 Clear-to-send, F…
  88 2.023539   192.168.1.7            192.168.1.2            UDP      1088 49153 → 20  Len=…
  89 2.032750                          00:00:00_00:00:07 (…  802.11     14 Acknowledgement,…

        .... .0.. = More Fragments: This is the last fragment
        .... 0... = Retry: Frame is not being retransmitted
        ...0 .... = PWR MGT: STA will stay up
        ..0. .... = More Data: No data buffered
        .0.. .... = Protected flag: Data is not protected
        1... .... = Order flag: Strictly ordered
    .010 0100 0000 0100 = Duration: 9220 microseconds
    Receiver address: 00:00:00_00:00:07 (00:00:00:00:00:07)
```

Figure 2: Network Allocation Vector from CTS frame

In the image above it can be seen that for the selected case the duration is 9220 microseconds. This is the network allocation vector information.

Conclusion:-

From this setup we learned how the Ad-hoc System can be implemented using NS-3 and how to implement RTS/CTS into it. We then observed that when RTS/CTS was implemented no collisions happened.

Task 2:-

<u>Experimental Setup:</u>

The setup parameters are the same as task 1, except that instead of Ad-hoc mode, the Wireless LAN is now operating in Infrastructure mode with the presence of an AP. The SSID to be used is PNET.

Changes made to the code from Node 1:

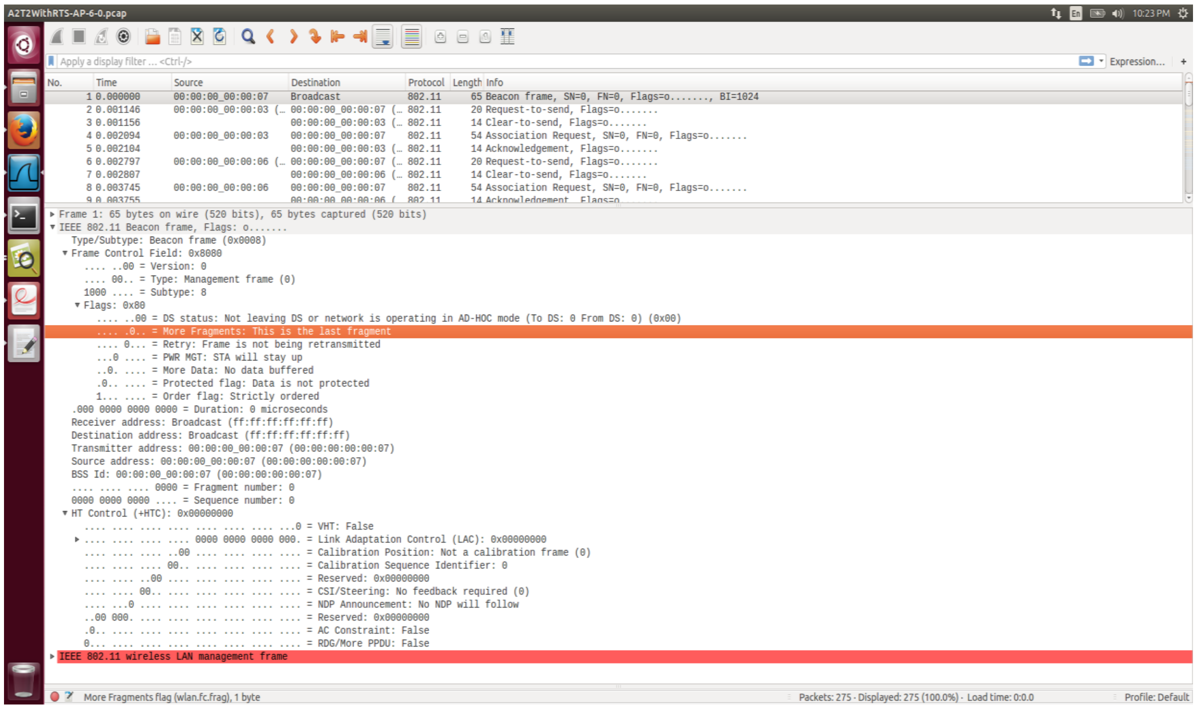
- The access point was added by the following code:
  NodeContainer wifiapnode;
  wifiapnode.Create(1);

- The SSID was set to PNET, and six nodes were set to infrastructure mode and an access point also installed. This was achieved by the following code:
  ```
  Ssid ssid = Ssid ("Pnet");
  mac.SetType ("ns3::StaWifiMac","Ssid",SsidValue (ssid));
  NetDeviceContainer adDevices;
  adDevices = wifi.Install (phy, mac, wifinodes);

  mac.SetType ("ns3::ApWifiMac","Ssid",SsidValue (ssid));
  NetDeviceContainer apDevices;
  apDevices = wifi.Install (phy, mac, wifiapnode);
  ```

<u>Task 2 Answers:</u>

1. The access point sends out beacon frames when the network starts operating. These beacon frames are sent out so as to inform the nodes of its presence. After this the nodes send out association requests to connect to the Access point. These association requests are responded to by the access point confirming the association of the nodes with AP. The transmission then starts between the access point and the nodes. The AP sends out beacon frames in between the transmission as well as after the end of the transmission. The periodical beacon frames that are sent by the AP contains information about the access point, security and about the buffered traffic of different nodes.

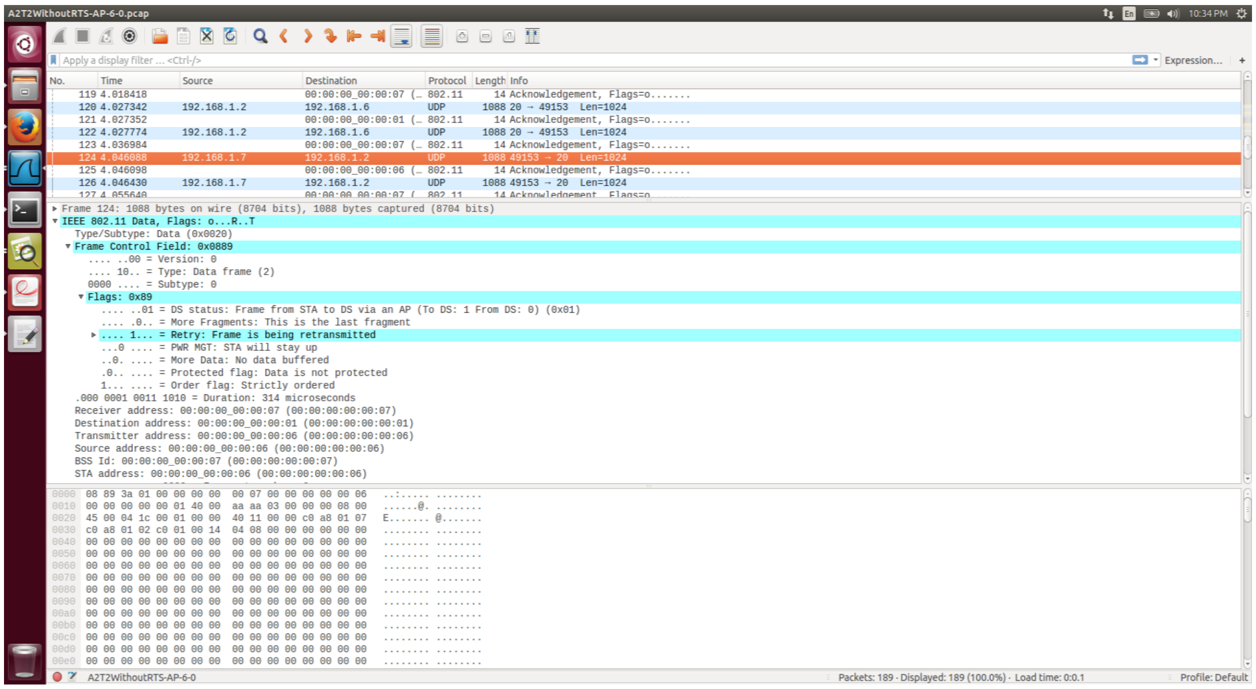
2. From the figure below the important parameters of beacon frame are:
i) Receiver Address
ii) Transmitter Address
iii) Source Address
iv) Frame retransmission information (Whether the frame is retransmitted or not)
v) Whether data is being buffered or not



```
A2T2WithRTS-AP-6-0.pcap

No.  Time      Source                 Destination            Protocol Length Info
   1 0.000000  00:00:00_00:00:07      Broadcast              802.11    65 Beacon frame, SN=0, FN=0, Flags=o......., BI=1024
   2 0.001146  00:00:00_00:00:03 (…   00:00:00_00:00:07 (…   802.11    20 Request-to-send, Flags=o.......
   3 0.001156                         00:00:00_00:00:03 (…   802.11    14 Clear-to-send, Flags=o.......
   4 0.002094  00:00:00_00:00:03      00:00:00_00:00:07      802.11    54 Association Request, SN=0, FN=0, Flags=o.......
   5 0.002104                         00:00:00_00:00:03 (…   802.11    14 Acknowledgement, Flags=o.......
   6 0.002797  00:00:00_00:00:06 (…   00:00:00_00:00:07 (…   802.11    20 Request-to-send, Flags=o.......
   7 0.002807                         00:00:00_00:00:06 (…   802.11    14 Clear-to-send, Flags=o.......
   8 0.003745  00:00:00_00:00:06      00:00:00_00:00:07      802.11    54 Association Request, SN=0, FN=0, Flags=o.......
   9 0.003755                         00:00:00_00:00:06 (…   802.11    14 Acknowledgement, Flags=o.......

▸ Frame 1: 65 bytes on wire (520 bits), 65 bytes captured (520 bits)
▾ IEEE 802.11 Beacon frame, Flags: o.......
    Type/Subtype: Beacon frame (0x0008)
  ▾ Frame Control Field: 0x8080
      .... ..00 = Version: 0
      .... 00.. = Type: Management frame (0)
      1000 .... = Subtype: 8
    ▾ Flags: 0x80
        .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
        .... .0.. = More Fragments: This is the last fragment
        .... 0... = Retry: Frame is not being retransmitted
        ...0 .... = PWR MGT: STA will stay up
        ..0. .... = More Data: No data buffered
        .0.. .... = Protected flag: Data is not protected
        1... .... = Order flag: Strictly ordered
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: 00:00:00_00:00:07 (00:00:00:00:00:07)
    Source address: 00:00:00_00:00:07 (00:00:00:00:00:07)
    BSS Id: 00:00:00_00:00:07 (00:00:00:00:00:07)
    .... .... .... 0000 = Fragment number: 0
    0000 0000 0000 .... = Sequence number: 0
  ▾ HT Control (+HTC): 0x00000000
      .... .... .... .... .... .... .... ...0 = VHT: False
    ▸ .... .... .... .... 0000 0000 0000 000. = Link Adaptation Control (LAC): 0x00000000
      .... .... .... ..00 .... .... .... .... = Calibration Position: Not a calibration frame (0)
      .... .... .... 00.. .... .... .... .... = Calibration Sequence Identifier: 0
      .... .... ..00 .... .... .... .... .... = Reserved: 0x00000000
      .... .... 00.. .... .... .... .... .... = CSI/Steering: No feedback required (0)
      .... ...0 .... .... .... .... .... .... = NDP Announcement: No NDP will follow
      ..00 000. .... .... .... .... .... .... = Reserved: 0x00000000
      .0.. .... .... .... .... .... .... .... = AC Constraint: False
      0... .... .... .... .... .... .... .... = RDG/More PPDU: False
▸ IEEE 802.11 wireless LAN management frame

More Fragments flag (wlan.fc.frag), 1 byte    Packets: 275 · Displayed: 275 (100.0%) · Load time: 0:0.0    Profile: Default
```

Figure 3: Parameters of Beacon Frame

3. Yes, there is a collision that is happening, as node 4 and node 5 send frames at the same time i.e 4s to the access point. Since there is no RTS/CTS there was a collision and it can be seen from the wireshark trace file that there was retransmission.



```
A2T2WithoutRTS-AP-6-0.pcap

No.  Time      Source         Destination            Protocol Length Info
 119 4.018418                 00:00:00_00:00:07 (…   802.11    14 Acknowledgement, Flags=o.......
 120 4.027342  192.168.1.2    192.168.1.6            UDP     1088 20 → 49153  Len=1024
 121 4.027352                 00:00:00_00:00:01 (…   802.11    14 Acknowledgement, Flags=o.......
 122 4.027774  192.168.1.2    192.168.1.6            UDP     1088 20 → 49153  Len=1024
 123 4.036984                 00:00:00_00:00:07 (…   802.11    14 Acknowledgement, Flags=o.......
 124 4.046088  192.168.1.7    192.168.1.2            UDP     1088 49153 → 20  Len=1024
 125 4.046098                 00:00:00_00:00:06 (…   802.11    14 Acknowledgement, Flags=o.......
 126 4.046430  192.168.1.7    192.168.1.2            UDP     1088 49153 → 20  Len=1024
 127 4.055640                 00:00:00_00:00:07 (…   802.11    14 Acknowledgement, Flags=o.......

▸ Frame 124: 1088 bytes on wire (8704 bits), 1088 bytes captured (8704 bits)
▾ IEEE 802.11 Data, Flags: o...R..T
    Type/Subtype: Data (0x0020)
  ▾ Frame Control Field: 0x0889
      .... ..00 = Version: 0
      .... 10.. = Type: Data frame (2)
      0000 .... = Subtype: 0
    ▾ Flags: 0x89
        .... ..01 = DS status: Frame from STA to DS via an AP (To DS: 1 From DS: 0) (0x01)
        .... .0.. = More Fragments: This is the last fragment
      ▸ .... 1... = Retry: Frame is being retransmitted
        ...0 .... = PWR MGT: STA will stay up
        ..0. .... = More Data: No data buffered
        .0.. .... = Protected flag: Data is not protected
        1... .... = Order flag: Strictly ordered
    .000 0001 0011 1010 = Duration: 314 microseconds
    Receiver address: 00:00:00_00:00:07 (00:00:00:00:00:07)
    Destination address: 00:00:00_00:00:01 (00:00:00:00:00:01)
    Transmitter address: 00:00:00_00:00:06 (00:00:00:00:00:06)
    Source address: 00:00:00_00:00:06 (00:00:00:00:00:06)
    BSS Id: 00:00:00_00:00:07 (00:00:00:00:00:07)
    STA address: 00:00:00_00:00:06 (00:00:00:00:00:06)

0000  08 89 3a 01 00 00 00 00  00 07 00 00 00 00 00 06   ..:..... ........
0010  00 00 00 00 00 01 40 00  aa aa 03 00 00 00 08 00   ......@. ........
0020  45 00 04 1c 00 01 00 00  40 11 00 00 c0 a8 01 07   E....... @.......
0030  c0 a8 01 02 c0 01 00 14  04 08 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........

A2T2WithoutRTS-AP-6-0    Packets: 189 · Displayed: 189 (100.0%) · Load time: 0:0.1    Profile: Default
```

Figure 4: Retransmission without RTS/CTS

4. After RTS/CTS was implemented no collisions were observed. We implement RTS/CTS to minimize the collisions by making an initial handshake. The RTS frame specifies how much data the node needs to transmit and the CTS frame specifies for how long the channel will be occupied. In case of WiFi Standard CTS is called Network Allocation Vector (NAV).

<u>Conclusion:-</u>

From this setup we learned how the Infrastructure mode for Wifi system can be implemented using NS-3 and how to implement RTS/CTS into it. We then observed that when RTS/CTS was implemented no collisions happened.

Group 4

Pulkit Hanswal

Roshan Baby Reji