CS6603: Advanced Topics in Wireless Networks
Assignment 2

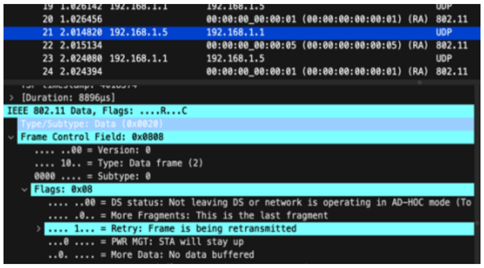
Task 1: Wireless LAN – Ad-hoc Mode

Experimental Setup:

I have used `NodeContainer` to create 6 nodes. And used `WifiHelper` to set standard to IEEE 801.11G using `WIFI_STANDARD_80211g` and to set Adaptive rate control to `AarfWifiManager` and added config to enable RTS/CTS using `ns3::WifiRemoteStationManager::RtsCtsThreshold` in order to enable it or disable it we just need to change `enableRts = false` this flag and as mentioned we are using default WIFI channel and PHY helper. By using `WifiMacHelper` we have defined our network is in Ad-hoc mode by setting mac type to `ns3::AdhocWifiMac` And created SSID using the string `SSID(“my-adhoc-network”)`. And Mobility of nodes are set using `MobilityHelper` with rectangular bound of 100m on each side. And the way we create Internet Stack, IP address assignment, Client and Server creation is similar to the Assignment 1. Finally, we have enabled Pcap for only Node 2 using “`wifiPhy.EnablePcap("third", devices.Get(2))`”.

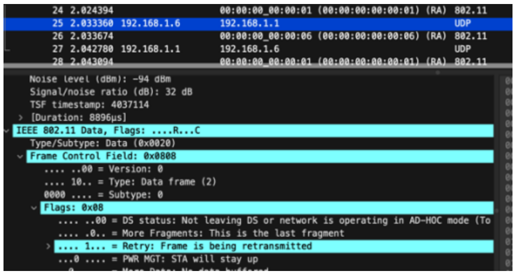
- **Are all the frames acknowledged? Explain why.**
  - Not all frames require acknowledgment, as some network protocols handle packet loss or duplication without ACKs. When a device sends an ARP request to resolve the MAC address of a specific IP address on the network, it does not wait for a response or acknowledgment before sending further packets. ARP operates as a simple broadcast-based protocol, where an ARP request is broadcast to all devices on the network, and the device with the matching IP address responds with an ARP reply containing its MAC address. Since ARP operates on the data link layer of the OSI model, it is not designed to provide reliable packet delivery, and it does not include any error detection or correction mechanisms. Therefore, all frames are not acknowledged by the receiver.

- **Are there any collisions in the network? Explain why. How have you reached this conclusion?**
  - Yes, there are collisions in this network. The collisions are in between frame 21 and frame 25 as these packets are being retransmitted, we can notify that the packets 21 and 25 are in collision.



```
19 1.026142 192.168.1.1    192.168.1.5                                   UDP
20 1.026456                00:00:00_00:00:01 (00:00:00:00:00:01) (RA)    802.11
21 2.014820 192.168.1.5    192.168.1.1                                   UDP
22 2.015134                00:00:00_00:00:05 (00:00:00:00:00:05) (RA)    802.11
23 2.024080 192.168.1.1    192.168.1.5                                   UDP
24 2.024394                00:00:00_00:00:01 (00:00:00:00:00:01) (RA)    802.11

> [Duration: 8896µs]
IEEE 802.11 Data, Flags: ....R...C
  Type/Subtype: Data (0x0020)
v Frame Control Field: 0x0808
    .... ..00 = Version: 0
    .... 10.. = Type: Data frame (2)
    0000 .... = Subtype: 0
  v Flags: 0x08
      .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To
      .... .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
```



```
24 2.024394                00:00:00_00:00:01 (00:00:00:00:00:01) (RA)    802.11
25 2.033360 192.168.1.6    192.168.1.1                                   UDP
26 2.033674                00:00:00_00:00:06 (00:00:00:00:00:06) (RA)    802.11
27 2.042780 192.168.1.1    192.168.1.6                                   UDP
28 2.043094                00:00:00_00:00:01 (00:00:00:00:00:01) (RA)    802.11

    Noise level (dBm): -94 dBm
    Signal/noise ratio (dB): 32 dB
    TSF timestamp: 4037114
  > [Duration: 8896µs]
v IEEE 802.11 Data, Flags: ....R...C
    Type/Subtype: Data (0x0020)
  v Frame Control Field: 0x0808
      .... ..00 = Version: 0
      .... 10.. = Type: Data frame (2)
      0000 .... = Subtype: 0
    v Flags: 0x08
        .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To
        .... .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
```

- **How can you force the nodes to utilize the RTS/CTS handshake procedure seen in class? What is the reasoning behind this procedure?**
  - Yes, using the “`ns3::WifiRemoteStationManager::RtsCtsThreshold`” value we can force the nodes to utilize the RTS`/CTS handshake procedure by setting the threshold value to zero. Suppose two or more devices are within range of a wireless access point but not in range with each other. When one device transmits data to the access point, another device may not detect the transmission. It may attempt to transmit data simultaneously, causing a collision. This is the hidden terminal problem. To mitigate the hidden terminal issue, it is recommended to use carrier sensing multiple access with collision avoidance (CSMA/CA) protocols. In CSMA/CA, devices listen for a clear channel before transmitting data to avoid collisions. The transmitting device sends a request to transmit (RTS) message to the access point, which then sends a clear-to-send (CTS) message to all other devices in the network. This allows other devices to defer their transmission until the channel is clear.

Force the utilization of RTS/CTS in the network:

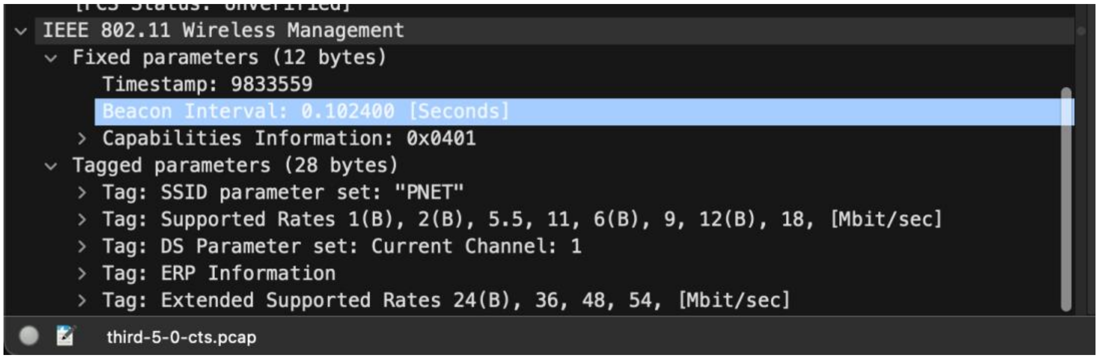
- Are there any collisions now?
  - After utilizing the RTS/CTS handshake procedure, there are no collisions.
- Which is the benefit or RTS/CTS?
  - RTS/CTS helps in reducing collisions happens due to hidden nodes and due to some other parameters.
- Where can you find the Network Allocation Vector information?
  - The Network Allocation Vector information is found in the IEEE 802.11 radio information, in the duration field and frame field respectively. And this helps in setting the medium busy for the specified time to avoid incoming requests and to avoid collisions.

Task 2: Wireless LAN – Infrastructure Mode

Experimental Setup:

The setup is same as previous one but in here we introduce Access Point Node. In the mac layer, `SSID` name is set to `PNET` and type is set to `StaWifiMac` to install wifi nodes and `ApWifiMac` to install AP node and in Mobility we used `ns3::ConstantPositionMobilityModel` for AP node as it stays in one place. And rest every thing is same as previous task. Finally, we enabled pcap for node 5 using `wifiPhy.EnablePcap("third", staDevices.Get(5));`

- Explain the behavior of the AP. What is happening since the very first moment the network starts operating?
  - As soon as the network is operational, the AP broadcasts its existence periodically by emitting a beacon signal. This signal includes the network's name (SSID), security options, and other configuration information.This beacon can be detected by devices in the AP's coverage area, and they can connect to the network using the data it carries. When a device joins the network, it can communicate with other networked devices using the AP.
- Take a look to a beacon frame. Which are the most relevant parameters defined in it?
  - There are two parameters defined in a beacon frame those include fixed parameters and tagged parameters. Fixed parameters include timestamp, beacon interval and capabilities information whereas the tagged parameters include SSID, supported rates, DS parameter, ERP Information and Extended Supported rates.



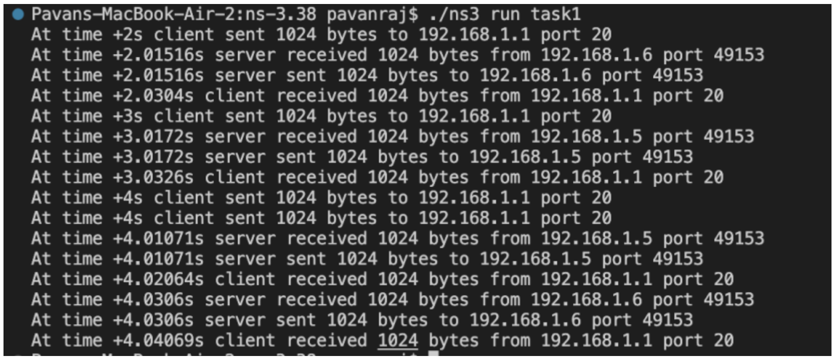
```
    [FCS Status: Unverified]
v IEEE 802.11 Wireless Management
  v Fixed parameters (12 bytes)
      Timestamp: 9833559
      Beacon Interval: 0.102400 [Seconds]
    > Capabilities Information: 0x0401
  v Tagged parameters (28 bytes)
    > Tag: SSID parameter set: "PNET"
    > Tag: Supported Rates 1(B), 2(B), 5.5, 11, 6(B), 9, 12(B), 18, [Mbit/sec]
    > Tag: DS Parameter set: Current Channel: 1
    > Tag: ERP Information
    > Tag: Extended Supported Rates 24(B), 36, 48, 54, [Mbit/sec]
```

third-5-0-cts.pcap

- Are there any collisions in the network? When are these collisions happening?
  - Yes there are collisions in the network.The collisions are between multiple frames which include frames 4, 8, 10, 16, 18, 19, 21, 23, 97,103, 125.This is because the frames are trying to communicate simultaneously.
- As in Task 1, force the utilization of the handshaking process and repeat the simulation. Are there any collisions now? Explain why?
  - There are no collisions in the network as the handshaking process is involved in the simulation.

Output:

As we are not changing the packets that we are sending to the server, the output trace is same for every simulation.



```
Pavans-MacBook-Air-2:ns-3.38 pavanraj$ ./ns3 run task1
At time +2s client sent 1024 bytes to 192.168.1.1 port 20
At time +2.01516s server received 1024 bytes from 192.168.1.6 port 49153
At time +2.01516s server sent 1024 bytes to 192.168.1.6 port 49153
At time +2.0304s client received 1024 bytes from 192.168.1.1 port 20
At time +3s client sent 1024 bytes to 192.168.1.1 port 20
At time +3.0172s server received 1024 bytes from 192.168.1.5 port 49153
At time +3.0172s server sent 1024 bytes to 192.168.1.5 port 49153
At time +3.0326s client received 1024 bytes from 192.168.1.1 port 20
At time +4s client sent 1024 bytes to 192.168.1.1 port 20
At time +4s client sent 1024 bytes to 192.168.1.1 port 20
At time +4.01071s server received 1024 bytes from 192.168.1.5 port 49153
At time +4.01071s server sent 1024 bytes to 192.168.1.5 port 49153
At time +4.02064s client received 1024 bytes from 192.168.1.1 port 20
At time +4.0306s server received 1024 bytes from 192.168.1.6 port 49153
At time +4.0306s server sent 1024 bytes to 192.168.1.6 port 49153
At time +4.04069s client received 1024 bytes from 192.168.1.1 port 20
```