# Report: Performance comparison of wireless protocol

# 802.11ax vs 802.11ac vs 802.11n

14.1.2023

Realized By: Halladj Hamza

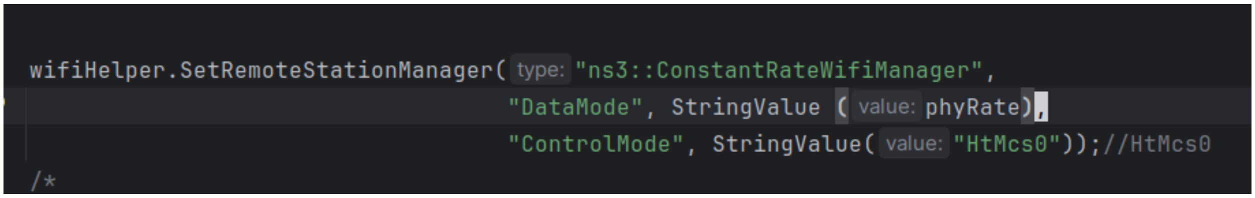
## 1. Abstract

When comparing IEEE 802.11ax to IEEE 802.11ac, which both operate in the 5 GHz band, IEEE 802.11ax Mcs-10 performs better than IEEE 802.11ac Mcs-9 and IEEE 802.11n Mcs-7, especially when there are a lot of customers. IEEE 802.11ax has a slightly longer initial latency, but after 0.5 milliseconds, it stabilizes more rapidly and maintains a greater throughput than IEEE 802.11ac & IEEE 802.11n, which have a diminishing throughput. It's crucial to recognize that the improvements seen in IEEE 802.11ax build upon the foundation laid by previous standards like IEEE 802.11n, which introduced features such as Multiple Input Multiple Output (MIMO) technology, contributing to improved data rates and reliability in wireless communication.

## 2. Parameter Settings

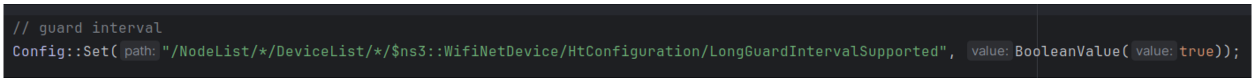
| | **802.11n** | **802.11ac** | **802.11ax** |
|---|---|---|---|
| **Modulation and Coding** | QAM-64 | QAM-256 | QAM-1024 |
| **Guard Interval** | 800 ns | 800 ns | 800 ns |
| **Channel Width** | 40 MHz | 40 MHz | 40 MHz |
| **Frequency** | 5 GHz | 5 GHz | 5 GHz |
| **Spatial Stream** | 1 | 1 | 1 |
| **Payload Size** | 1448 byte | 1448 byte | 1448 byte |
| **Client number** | 2-128 | 2-128 | 2-128 |
| **Simulation time** | 2 s | 2 s | 2 s |

- **Modulation and  Coding:** We used different Modulation & Coding schemes to measure the effect they have over the **Throughput** in Tcp traffic where we used the following values for the variable ***phyRate:*** *HeMcs10 ⇒ QAM-1024, VhtMcs9 ⇒ QAM-256,    HtMcs7 ⇒ QAM-64*



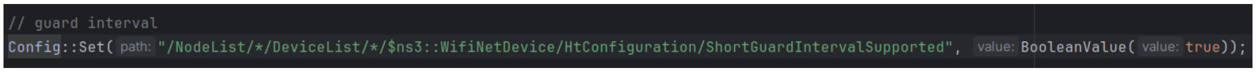
```
wifiHelper.SetRemoteStationManager( type: "ns3::ConstantRateWifiManager",
                                    "DataMode", StringValue ( value: phyRate),
                                    "ControlMode", StringValue( value: "HtMcs0"));//HtMcs0
/*
```

- **Guard Interval:** We used guard interval as 800 ns.
  This is for 802.11n and 802.11ac



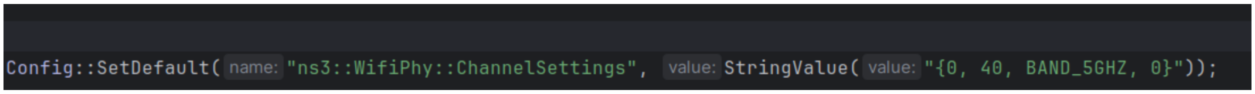
```
// guard interval
Config::Set( path: "/NodeList/*/DeviceList/*/$ns3::WifiNetDevice/HtConfiguration/LongGuardIntervalSupported",  value: BooleanValue( value: true));
```

  And this is for 802.11ax



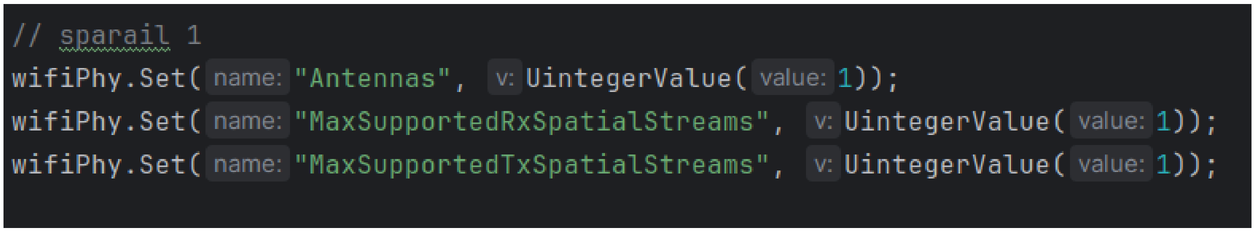
```
// guard interval
Config::Set( path: "/NodeList/*/DeviceList/*/$ns3::WifiNetDevice/HtConfiguration/ShortGuardIntervalSupported",  value: BooleanValue( value: true));
```

- **Channel Width & Frequency:** to have a Consistent study we must use 40 MHz sized channels because 802.11n only has 20,40 MHz sized channels we picked the 40 MHz to have the biggest throughput.



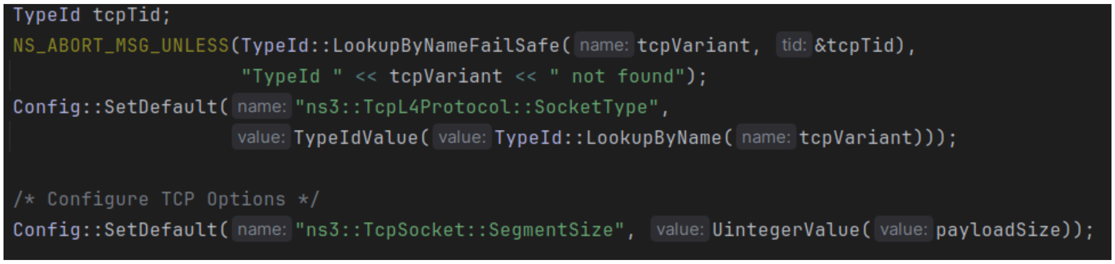
```
Config::SetDefault( name: "ns3::WifiPhy::ChannelSettings",  value: StringValue( value: "{0, 40, BAND_5GHZ, 0}"));
```

- **Spatial Stream:** we would use only one spatial stream with MIMO.



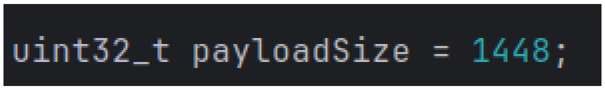
```
// sparail 1
wifiPhy.Set( name: "Antennas",  v: UintegerValue( value: 1));
wifiPhy.Set( name: "MaxSupportedRxSpatialStreams",  v: UintegerValue( value: 1));
wifiPhy.Set( name: "MaxSupportedTxSpatialStreams",  v: UintegerValue( value: 1));
```

- **Payload Size:** we are using the payload size 1448 to test the TCP `NewReno` Throughput.



```
TypeId tcpTid;
NS_ABORT_MSG_UNLESS(TypeId::LookupByNameFailSafe( name: tcpVariant,  tid: &tcpTid),
                    "TypeId " << tcpVariant << " not found");
Config::SetDefault( name: "ns3::TcpL4Protocol::SocketType",
                    value: TypeIdValue( value: TypeId::LookupByName( name: tcpVariant)));

/* Configure TCP Options */
Config::SetDefault( name: "ns3::TcpSocket::SegmentSize",  value: UintegerValue( value: payloadSize));
```

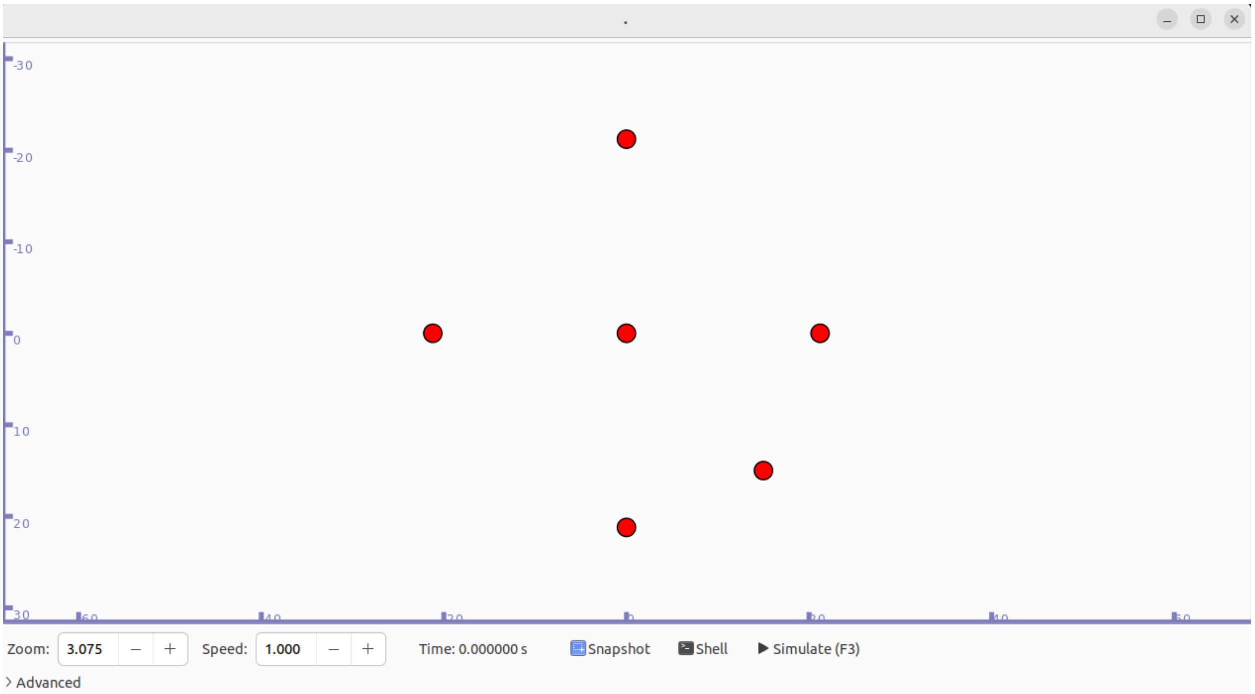


```
uint32_t payloadSize = 1448;
```

- **Client number & Simulation time:** we will be using 2-128 clients and the simulation time is 2 s.

# 3. Scenarios:

We used ns3-40 to simulate our examples and reach the results reliably and effectively. We used a **Start Topology** with **n** clients and an access point(AP). We relied on the TCP version New-Reno to have a higher reliability compared to regular TCP. We ran the simulation with the Mentioned configurations above, in order to find the maximum value of throughput of each protocol and compare them.



Zoom: 3.075 – +

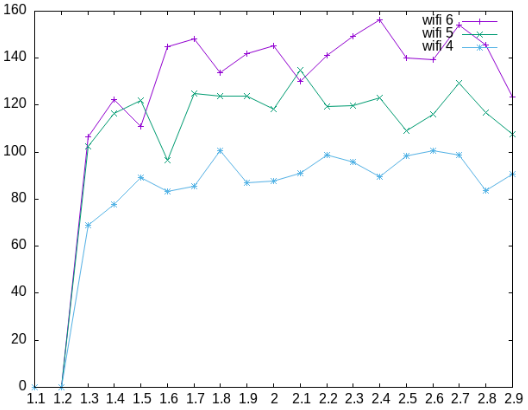
Speed: 1.000 – +

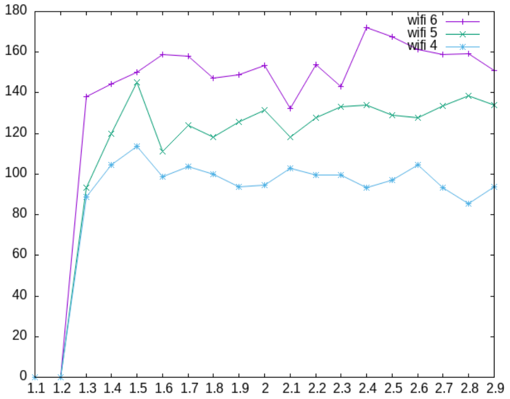
Time: 0.000000 s

# 4. RESULT AND DISCUSSION

## **A.** Number of clients: 2 & 4

- with 2 and 4 clients, IEEE 802.11ax 802.11ac and 802.11n exhibited swift response times and robust throughput. Notably, the bit rate of IEEE 802.11ax surpassed that of IEEE 802.11ac.

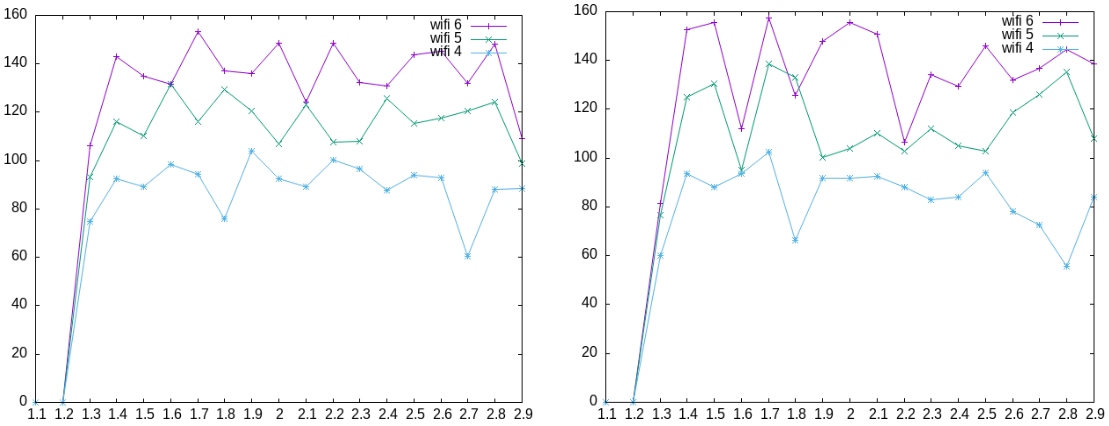




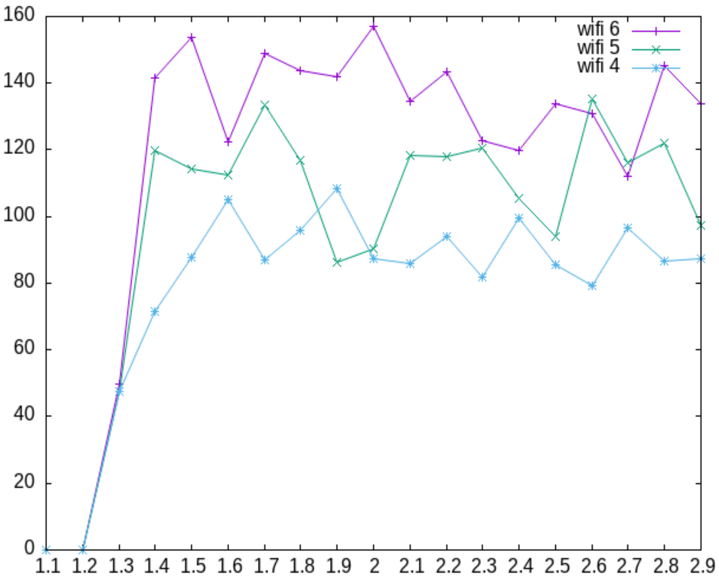
## **B.** Number of clients: 8, 16 & 32:

- *For simulations with 8, 16, and 32 clients,* 802.11ax 802.11ac and 802.11n *exhibited similar response times with a slight delay. They demonstrated good throughput, with the bit rate of IEEE 802.11ax consistently surpassing that of IEEE 802.11ac. This suggests that the MIMO implementation of 802.11ax is more effective than 802.11ac.*

- In the 32 clients, a dip occurred in the 802.11ac around the *1.8 - 1.9 s* mark where it went below the 802.11n.



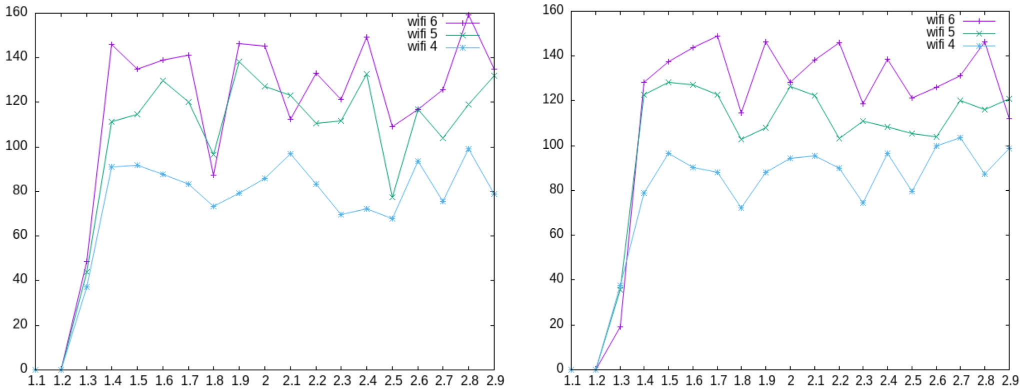
>> The one on the left is the  diagram for 8 clients and the one on the right represents the 16 clients.



>>32 client diagram.

## C. Number of clients: 64 & 128:

- For simulations with a larger client number (64, 128), the results indicate that IEEE 802.11ac exhibits a faster response than IEEE 802.11ax. In this context, it's valuable to bring IEEE 802.11n into the comparison. Despite its earlier generation, IEEE 802.11n might demonstrate competitive performance, especially in scenarios with a larger number of clients.



>> The one on the left is the diagram for 64 clients and the one on the right represents the 128 clients.

- IEEE 802.11ax, IEEE 802.11ac, and IEEE 802.11n all showed quick reaction times for lower client numbers (2 and 4), with 802.11ax displaying a higher bit rate. Throughout the two-second simulation, the throughput for each of the three procedures stayed constant. In situations including eight, sixteen, and thirty-two medium-sized client numbers, all three protocols demonstrated comparable response times and acceptable throughput beginning at 0.3 seconds. IEEE 802.11ax's MIMO implementation proved to be effective as it constantly outperformed IEEE 802.11ac and IEEE 802.11n in bit rate, even though the simulation was stable throughout.

## 5. CONCLUSION

IEEE 802.11ax emerges as the standout performer, which outperforms both IEEE 802.11ac and IEEE 802.11n in terms of bit rate and continues to have a greater throughput until the simulation is finished. This result demonstrates the resilience of IEEE 802.11ax, whose cutting-edge capabilities—such as OFDMA and enhanced MIMO—have proven crucial in providing better performance, particularly in situations with greater client count.