Dibyajyoti Deka (CSE23011)
Khyati Chowdhury (CSE23027)

---

Laboratory Assignment #1: Performance Analysis of IEEE 802.11 MAC Protocol Using Mininet- WiFi

1. Part 1 : Hidden Terminal Problem Simulation

## Network Topology and Configuration

This network simulation is designed to create a simple topology with three stations and one access point, managed by a software-defined network controller. The following steps outline the network setup, configuration, and packet capture process.

(a) **Network Topology**:

i. The setup includes three stations, `sta1`, `sta2`, and `sta3`, each with unique MAC and IP addresses.

ii. An access point, `ap1`, is configured to broadcast an SSID labeled "handover", operating on WiFi channel 1 in mode `g`.

(b) **Node and Access Point Details**:

i. The network contains four nodes: three stations and one access point.

ii. Node positions are as follows:

A. `sta1`: Position $(30, 70, 0)$. This station is out of range from `sta3`, but within range of `sta2`.

B. `sta2`: Position $(50, 70, 0)$.

C. `sta3`: Position $(70, 70, 0)$. This station is out of range from `sta1`, but within range of `sta2`.

D. `ap1`: Position $(50, 65, 0)$.

(c) **Medium Access and Propagation Model**:

i. Medium access uses `wmediumd` with an interference model which simulates realistic wireless interference.

ii. The `logDistance` propagation model is applied in our simulation with an exponent of 5. This models the worst case for signal attenuation over distance.

(d) **Packet Capture**:

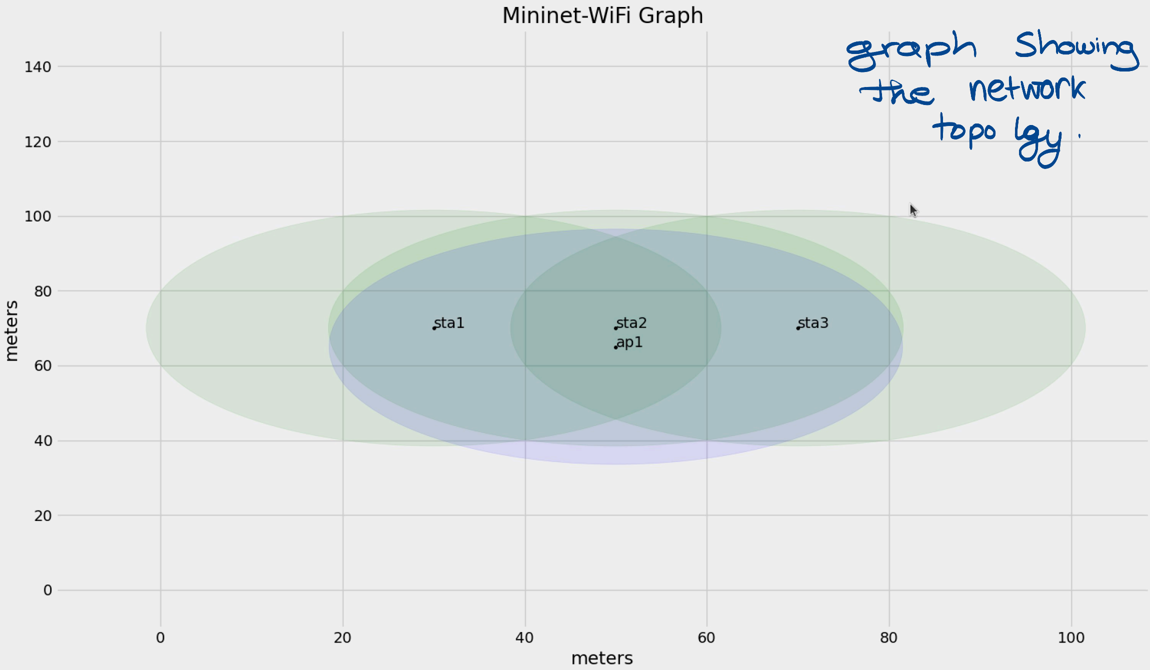
i. Packet data is captured on the virtual `hwsim0` interface.

ii. `tcpdump` is used to capture packets and save them to a `.pcap` file for later analysis.

(e) **Traffic Simulation Using iPerf**:

i. UDP traffic is generated for testing purposes, with `sta2` running iPerf servers on ports 5565 and 5566.

ii. `sta1` and `sta3` send UDP traffic to `sta2` at bandwidths varying from 1 Mbps to 100 Mbps for 15 seconds through 14 iterations of our simulation to observe packet flow, delay, and packet loss.

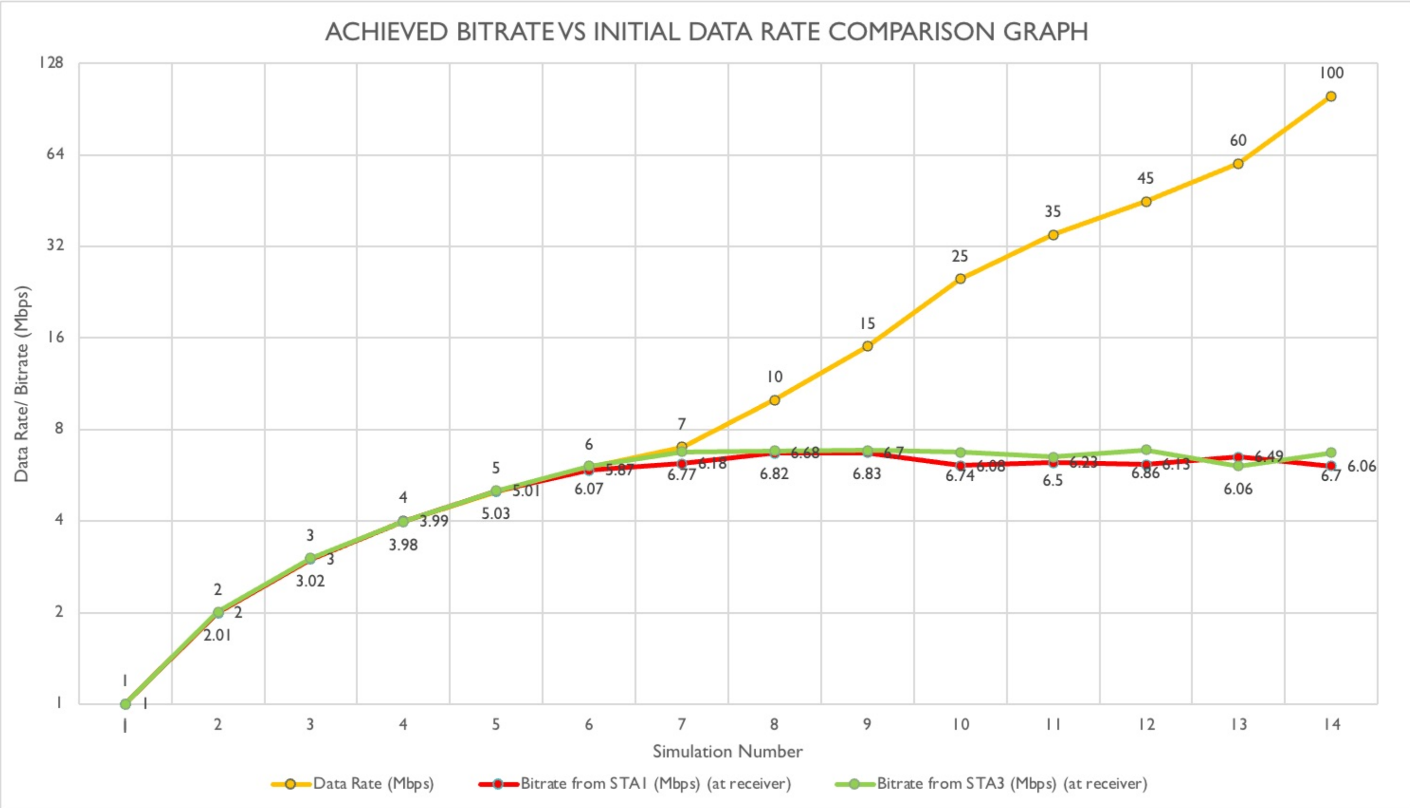



graph Showing the network topo lgy.

# Results from Multiple Simulations

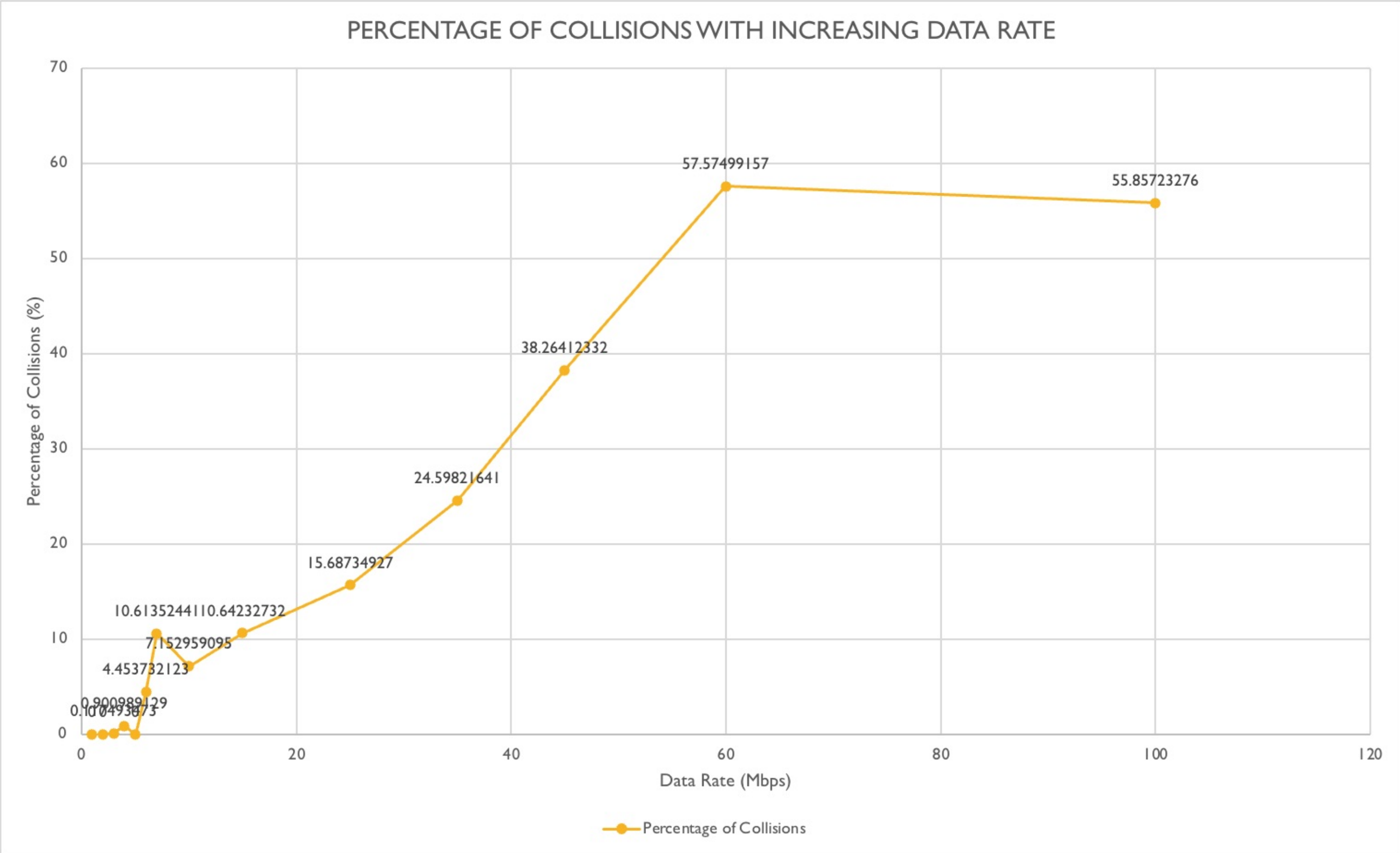
| Simulation No. | Data Rate (Mbps) | Time (seconds) | Bitrate from STA1 (Mbps) (at receiver) | Bitrate from STA3 (Mbps) (at receiver) | Packets lost through collisions from STA1 | Packets lost through collisions from STA3 | Total packets from STA1 | Total packets from STA3 | Total Collisions | Total Packets | Percentage of Collisions |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 15 | 1 | 1 | 0 | 0 | 1279 | 1279 | 0 | 2558 | 0 |
| 2 | 2 | 15 | 2 | 2.01 | 0 | 0 | 2554 | 2554 | 0 | 5108 | 0 |
| 3 | 3 | 15 | 3 | 3.02 | 0 | 9 | 3830 | 3830 | 9 | 7660 | 0.117493473 |
| 4 | 4 | 15 | 3.99 | 3.98 | 36 | 56 | 5105 | 5106 | 92 | 10211 | 0.900989129 |
| 5 | 5 | 15 | 5.01 | 5.03 | 0 | 0 | 6381 | 6381 | 0 | 12762 | 0 |
| 6 | 6 | 15 | 5.87 | 6.07 | 203 | 479 | 7656 | 7657 | 682 | 15313 | 4.453732123 |
| 7 | 7 | 15 | 6.18 | 6.77 | 987 | 909 | 8932 | 8932 | 1896 | 17864 | 10.61352441 |
| 8 | 10 | 15 | 6.68 | 6.82 | 555 | 760 | 9155 | 9229 | 1315 | 18384 | 7.152959095 |
| 9 | 15 | 15 | 6.7 | 6.83 | 627 | 1396 | 9245 | 9764 | 2023 | 19009 | 10.64232732 |
| 10 | 25 | 15 | 6.08 | 6.74 | 187 | 2675 | 7902 | 10342 | 2862 | 18244 | 15.68734927 |
| 11 | 35 | 15 | 6.23 | 6.5 | 1003 | 4155 | 9009 | 11960 | 5158 | 20969 | 24.59821641 |
| 12 | 45 | 15 | 6.13 | 6.86 | 866 | 8542 | 8576 | 16011 | 9408 | 24587 | 38.26412332 |
| 13 | 60 | 15 | 6.49 | 6.06 | 13300 | 7199 | 20635 | 14969 | 20499 | 35604 | 57.57499157 |
| 14 | 100 | 15 | 6.06 | 6.7 | 527 | 18112 | 8013 | 25356 | 18639 | 33369 | 55.85723276 |

| RESULT 1 | | | |
|---|---|---|---|
| ACHIEVED BITRATE VS INITIAL DATA RATE COMPARISON GRAPH | | | |
| Simulation No. | Data Rate (Mbps) | Bitrate from STA1 (Mbps) (at receiver) | Bitrate from STA3 (Mbps) (at receiver) |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2.01 |
| 3 | 3 | 3 | 3.02 |
| 4 | 4 | 3.99 | 3.98 |
| 5 | 5 | 5.01 | 5.03 |
| 6 | 6 | 5.87 | 6.07 |
| 7 | 7 | 6.18 | 6.77 |
| 8 | 10 | 6.68 | 6.82 |
| 9 | 15 | 6.7 | 6.83 |
| 10 | 25 | 6.08 | 6.74 |
| 11 | 35 | 6.23 | 6.5 |
| 12 | 45 | 6.13 | 6.86 |
| 13 | 60 | 6.49 | 6.06 |
| 14 | 100 | 6.06 | 6.7 |




As we can see from Result 1 the bitrate from both stations starts to cap at around 6.8 Mbps as much as we increase the data rate. The graphs show the both the data rate and the throughput/bit rate are matching till about 7 mbps, but deviating there after.

| RESULT 2 | | | | |
|---|---|---|---|---|
| PERCENTAGE OF COLLISIONS WITH INCREASING DATA RATE | | | | |
| Simulation No. | Data Rate (Mbps) | Total Collisions | Total Packets | Percentage of Collisions |
| 1 | 1 | 0 | 2558 | 0 |
| 2 | 2 | 0 | 5108 | 0 |
| 3 | 3 | 9 | 7660 | 0.117493473 |
| 4 | 4 | 92 | 10211 | 0.900989129 |
| 5 | 5 | 0 | 12762 | 0 |
| 6 | 6 | 682 | 15313 | 4.453732123 |
| 7 | 7 | 1896 | 17864 | 10.61352441 |
| 8 | 10 | 1315 | 18384 | 7.152959095 |
| 9 | 15 | 2023 | 19009 | 10.64232732 |
| 10 | 25 | 2862 | 18244 | 15.68734927 |
| 11 | 35 | 5158 | 20969 | 24.59821641 |
| 12 | 45 | 9408 | 24587 | 38.26412332 |
| 13 | 60 | 20499 | 35604 | 57.57499157 |
| 14 | 100 | 18639 | 33369 | 55.85723276 |




Also, expanding on the results we can see that as data rate increases the number of packets lost due to collisions also increases.
Hence, we calculate the percentage of the collisions and plot it on a graph.

## Observations and Inferences on Results

(a) **Result 1: Achieved Bitrate vs. Initial Data Rate Comparison**

i. **Observation:** The simulation results reveal a clear trend in achieved bitrate as the initial data rate is varied. At lower data rates, such as 1 to 5 Mbps, the achieved bitrate from `STA1` and `STA3` aligns closely with the initial data rate, suggesting minimal interference and efficient transmission. However, as the initial data rate increases beyond this range, a saturation effect becomes evident. Achieved bitrates at both `STA1` and `STA3` level off around 6 to 7 Mbps, even when the initial data rate is set much higher, such as 45, 60, or 100 Mbps. This behavior indicates that the network reaches a limit in handling higher data rates, resulting in diminishing returns for data throughput.

ii. **Inference:** The observed bitrate plateau can be attributed to the hidden terminal problem in wireless networks, which becomes more pronounced as data rates increase. When multiple stations attempt to transmit simultaneously, particularly at higher data rates, hidden terminal nodes are unable to detect each other's transmissions. This lack of awareness leads to packet collisions, causing the network throughput to stabilize or even decline despite increased data rate settings. The hidden terminal problem thus limits effective throughput as the network struggles to handle simultaneous transmissions, especially at higher data rates where packet interference is more likely.

(b) **Result 2: Percentage of Collisions with Increasing Data Rate**

i. **Observation:** The percentage of collisions in the network remains negligible at the lowest data rates, with no collisions observed at 1 and 2 Mbps. However, as the data rate increases to 3 Mbps and beyond, collisions start to appear. At moderate data rates, such as 6 Mbps, the collision percentage is around 4.45%, indicating that contention within the network is increasing. This trend escalates significantly as the data rate reaches 15 Mbps and above, with collision percentages exceeding 10% and even surpassing 24.6% at 35 Mbps. At the highest data rates, such as 60 and 100 Mbps, more than half of the packets encounter collisions, with collision percentages exceeding 55%. This sharp rise in collisions demonstrates a breakdown in the network's ability to handle simultaneous high-rate transmissions.

ii. **Inference:** The rise in collision percentage with increasing data rate strongly suggests that the hidden terminal problem intensifies at higher transmission speeds. In scenarios where multiple nodes transmit concurrently without detecting each other, the likelihood of collision rises significantly, especially as nodes attempt to send data at higher rates. These hidden terminals contribute to frequent packet loss, as they interfere with each other's transmissions, leading to higher collision rates and lower effective throughput.

## Summary of Findings

(a) The network performs efficiently at lower data rates, with minimal packet collisions and high achieved bitrates. However, as data rates increase, the hidden terminal problem becomes a limiting factor, leading to higher collision rates and a saturation of achievable throughput.

(b) The observed plateau in achieved bitrate and the rising collision rates suggest that this network configuration may require specific adjustments to mitigate hidden terminal effects. Solutions such as the use of RTS/CTS protocols or adjustments in node positioning could improve performance for applications that demand reliable high-throughput transmission.

2. Part 2 : Power Saving Mode Analysis

## Network Topology and Configuration

(a) **Network Topology**

i. **Network Structure**: The simulation has four stations (`sta1`, `sta2`, `sta3`, and `sta4`) and one access point (`ap1`).

ii. **Node Roles**:

A. `sta1`: Power-Saving Mode (PSM)-enabled station.

B. `sta2`, `sta3`, `sta4`: Regular stations without PSM.

C. `ap1`: Access point with SSID `simplewifi` on channel 1 which is operating in the `g` mode.

(b) **Medium Used**

i. **Wireless Medium**: `wmediumd` with an interference mode to used mimic real world interference conditions.

(c) **Node Positions and Number**

i. **Station Positions**:

A. `sta1`: Position (0, 5, 0)

B. `sta2`: Position (15, 10, 0)

C. `sta3`: Position (20, 10, 0)

D. `sta4`: Position (15, 20, 0)

ii. **Access Point Position**:

A. `ap1`: Position (15, 15, 0)

iii. **Total Number of Nodes**: 5 (4 stations + 1 access point).

(d) **Power-Saving Mode (PSM)**

i. **PSM on `sta1`**: PSM is enabled for `sta1`, configured using the command `iw dev wlan0 set power_save on`.

(e) **Propagation Model**

i. **Model**: `logDistance` model with a path loss exponent of 5 generates a realistic propagation setting.

(f) **Packet Capture**

i. **Interface**: Packets are captured on the `hwsim0` interface.

ii. **Command**: Packet capture is started with the command `tcpdump -i hwsim0 -w filename.pcap`.

iii. **Capture File**: All captured packets are stored in a `.pcap` file

(g) **Traffic Generation**

i. **Traffic Type**: UDP traffic generated using `iperf`.

ii. **Flow**: Background traffic is created by `sta4` to other stations (`sta1`, `sta2`, `sta3`) with a varying data rate from 0.5 to 60 Mbps for a total of 11 simulations, packet size of 1024 bytes, and duration of 25 seconds.

(h) **Power Consumption and Packet Counts**

i. **Packet Counting**: Transmit (TX) and receive (RX) packet counts are obtained by us for each station before and after traffic generation to estimate power consumption which is then used to calculate active time of the station based on packet activity/duty cycles.

ii. **Commands Used**:

A. **TX Packets**: `cat /sys/class/net/{iface}/statistics/tx_packets`

B. **RX Packets**: `cat /sys/class/net/{iface}/statistics/rx_packets`

(i) **ICMP Latency Calculation**

i. **Objective**: To measure the impact of Power-Saving Mode (PSM) on ICMP latency.

ii. **Method**: Each station (`sta1`, `sta2`, `sta3`) sends 20 ICMP packets (ping requests) to `sta4`.

iii. **Command Used**:

A. The following command is executed on each station: `ping -c 20 <sta4_IP>`.

iv. **Data Analysis**: Latency values are accumulated for each ping request to `sta4`, we then calculate the average latency for every station. The differences in latency between PSM-enabled (`sta1`) and non-PSM stations are recorded by us to understand the effect of Power Saving Mode on ICMP response times.

(j) **Inter-Packet Latency Calculation from UDP Packets in PCAP File**

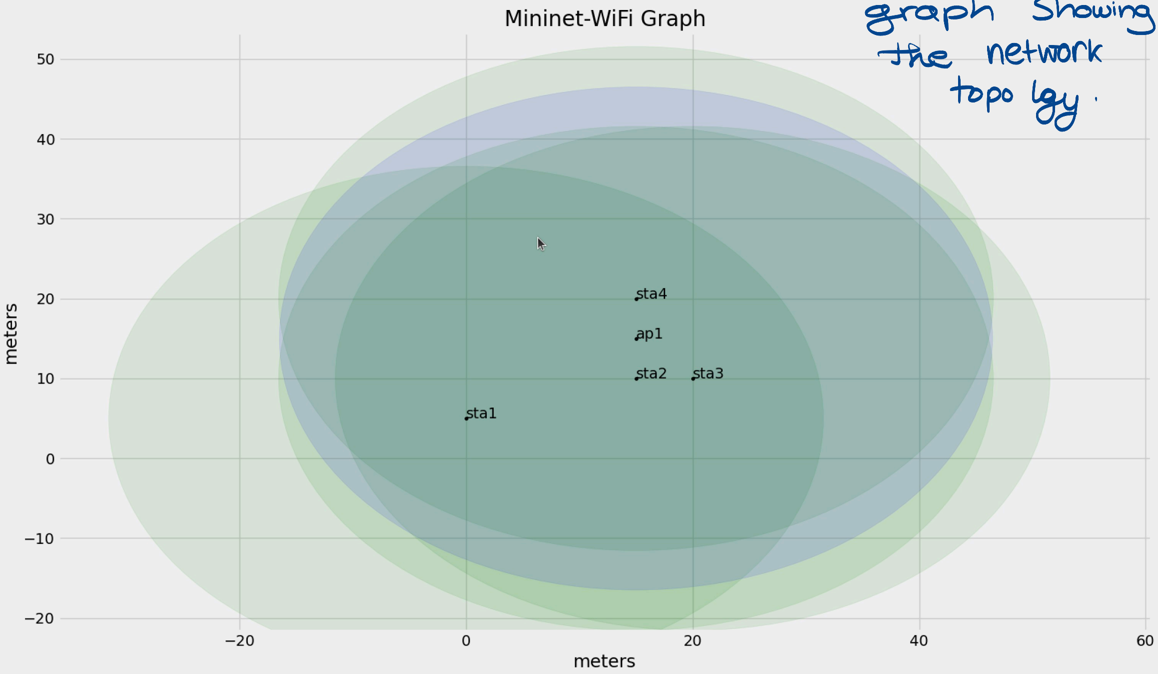
i. **Objective**: To calculate the inter-packet latency for UDP packets captured in the PCAP file.

ii. **Method**:

   A. **PCAP File Analysis**: The `.pcap` file, captured using `tcpdump`, is analyzed with a Python script using `scapy` to extract the timestamps of UDP packets.

   B. **Inter-Packet Latency**: Timestamps of consecutive UDP packets will be subtracted from each other to get times in between packet arrivals which essentially tell us the latency for each packet.

iii. **Data Analysis**: Inter-packet latencies are derived and examined for latency distribution, deviation, and mean delay in packet arrival time, especially with respect to the presence of Power Saving Mode on `sta1`.




graph Showing the network topo lgy.

# Results from Multiple Simulations

| Simulation No. | Data Rate (Mbps) | Duration (s) | STA1:ICMP Latency (ms) | STA1: Avg Inter Packet Latency (ms) | STA1: Max Inter Packet Latency (ms) | STA1: TX Packets | STA1: RX Packets | STA1: Duty Cycle (%) |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.5 | 25 | 1.55 | 8.1847 | 35.591 | 5 | 1542 | 33.31180017 |
| 2 | 5 | 25 | 60.051 | 0.8197 | 12.953 | 6 | 15258 | 33.3086021 |
| 3 | 10 | 25 | 56.707 | 0.5359 | 9.974 | 5 | 18862 | 32.99753397 |
| 4 | 15 | 25 | 248.731 | 0.5236 | 28.179 | 5 | 9494 | 32.51189376 |
| 5 | 20 | 25 | 49.853 | 0.3714 | 3815.074 | 12 | 13341 | 28.20897414 |
| 6 | 25 | 25 | 57.951 | 0.3788 | 48.492 | 4 | 5122 | 31.75370129 |
| 7 | 27 | 25 | 49.15 | 0.357 | 65.228 | 4 | 5370 | 31.94436189 |
| 8 | 30 | 25 | 55.915 | 0.243 | 5.402 | 6 | 9981 | 32.49495673 |
| 9 | 40 | 25 | 57.623 | 0.239 | 46.401 | 4 | 5401 | 31.26626945 |
| 10 | 50 | 25 | 60.97 | 0.3096 | 7114.338 | 4 | 5400 | 25.05563798 |
| 11 | 60 | 25 | 65.1 | 0.2139 | 25.26 | 4 | 1705 | 27.22638203 |

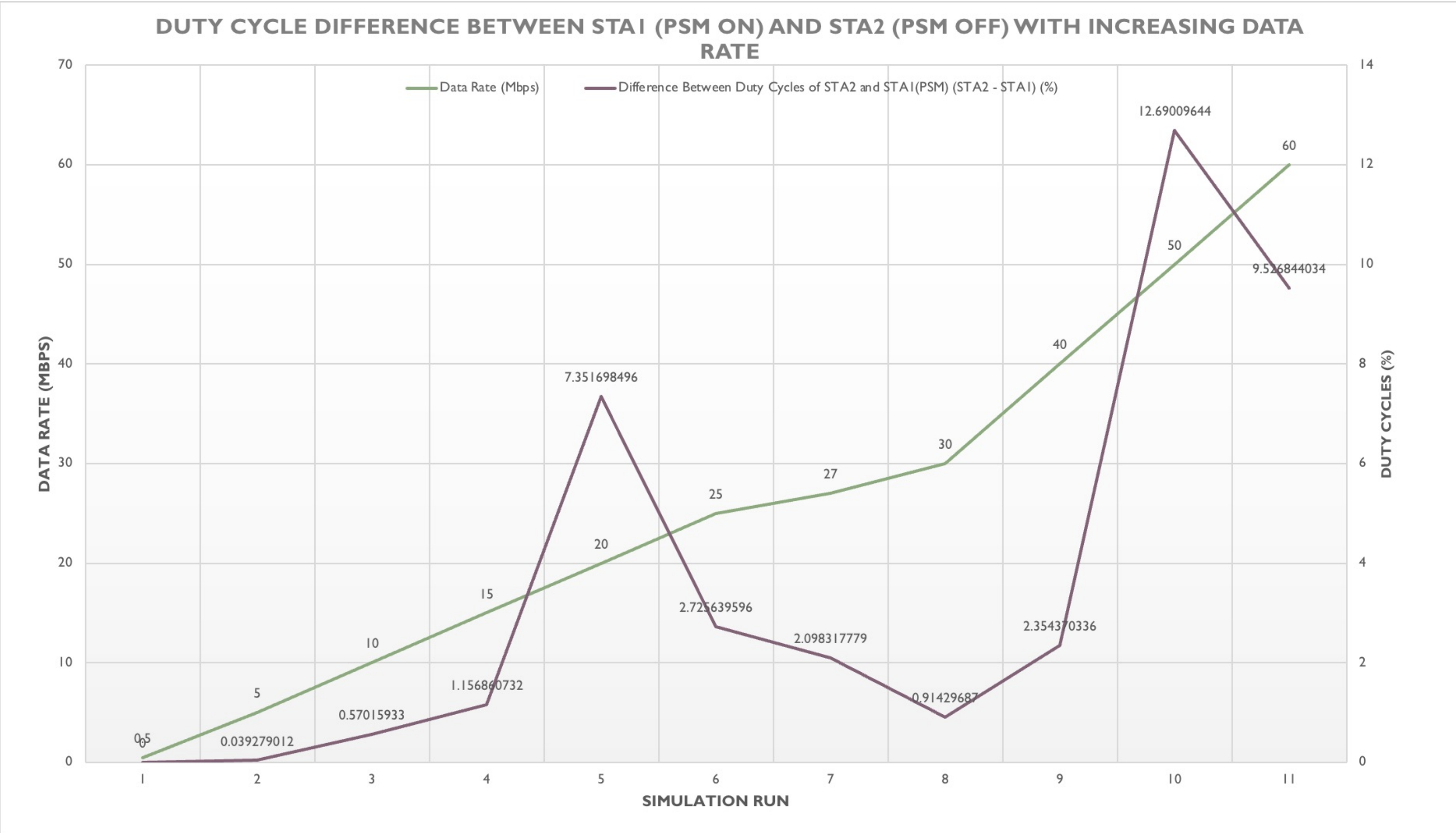
| Simulation No. | Data Rate (Mbps) | Duration (s) | STA2:ICMP Latency (ms) | STA2: Avg Inter Packet Latency (ms) | STA2: Max Inter Packet Latency (ms) | STA2: TX Packets | STA2: RX Packets | STA2: Duty Cycle (%) |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.5 | 25 | 1.6 | 8.1846 | 35.591 | 8 | 1539 | 33.31180017 |
| 2 | 5 | 25 | 2.102 | 0.8191 | 3.675 | 9 | 15273 | 33.34788112 |
| 3 | 10 | 25 | 1.278 | 0.5365 | 18.514 | 5 | 19188 | 33.5676933 |
| 4 | 15 | 25 | 1.242 | 0.5222 | 42.719 | 5 | 9832 | 33.66875449 |
| 5 | 20 | 25 | 1.248 | 0.352 | 1831.412 | 12 | 16821 | 35.56067264 |
| 6 | 25 | 25 | 1.362 | 0.3774 | 48.511 | 6 | 5560 | 34.47934089 |
| 7 | 27 | 25 | 1.239 | 0.3556 | 70.473 | 6 | 5721 | 34.04267966 |
| 8 | 30 | 25 | 1.255 | 0.2425 | 4.73 | 6 | 10262 | 33.4092536 |
| 9 | 40 | 25 | 1.623 | 0.2383 | 48.011 | 6 | 5806 | 33.62063979 |
| 10 | 50 | 25 | 1.643 | 0.2802 | 7103.572 | 6 | 8135 | 37.74573442 |
| 11 | 60 | 25 | 1.684 | 0.2134 | 26.75 | 6 | 2301 | 36.75322606 |

| Simulation No. | Data Rate (Mbps) | Duration (s) | STA3:ICMP Latency (ms) | STA3: Avg Inter Packet Latency (ms) | STA3: Max Inter Packet Latency (ms) | STA3: TX Packets | STA3: RX Packets | STA3: Duty Cycle (%) |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.5 | 25 | 1.3 | 8.1845 | 35.197 | 9 | 1541 | 33.37639966 |
| 2 | 5 | 25 | 1.253 | 0.8192 | 4.919 | 8 | 15272 | 33.34351678 |
| 3 | 10 | 25 | 1.138 | 0.5377 | 14.085 | 14 | 19103 | 33.43477272 |
| 4 | 15 | 25 | 1.236 | 0.5214 | 37.929 | 5 | 9876 | 33.81935175 |
| 5 | 20 | 25 | 1.745 | 0.3506 | 1831.871 | 19 | 17131 | 36.23035322 |
| 6 | 25 | 25 | 1.517 | 0.3779 | 48.421 | 6 | 5445 | 33.76695781 |
| 7 | 27 | 25 | 1.183 | 0.3556 | 51.887 | 6 | 5716 | 34.01295845 |
| 8 | 30 | 25 | 1.26 | 0.2419 | 4.636 | 6 | 10473 | 34.09578968 |
| 9 | 40 | 25 | 1.799 | 0.2378 | 48.786 | 6 | 6064 | 35.11309076 |
| 10 | 50 | 25 | 1.555 | 0.2791 | 7210.367 | 6 | 8017 | 37.1986276 |
| 11 | 60 | 25 | 1.576 | 0.2131 | 25.377 | 6 | 2255 | 36.02039191 |

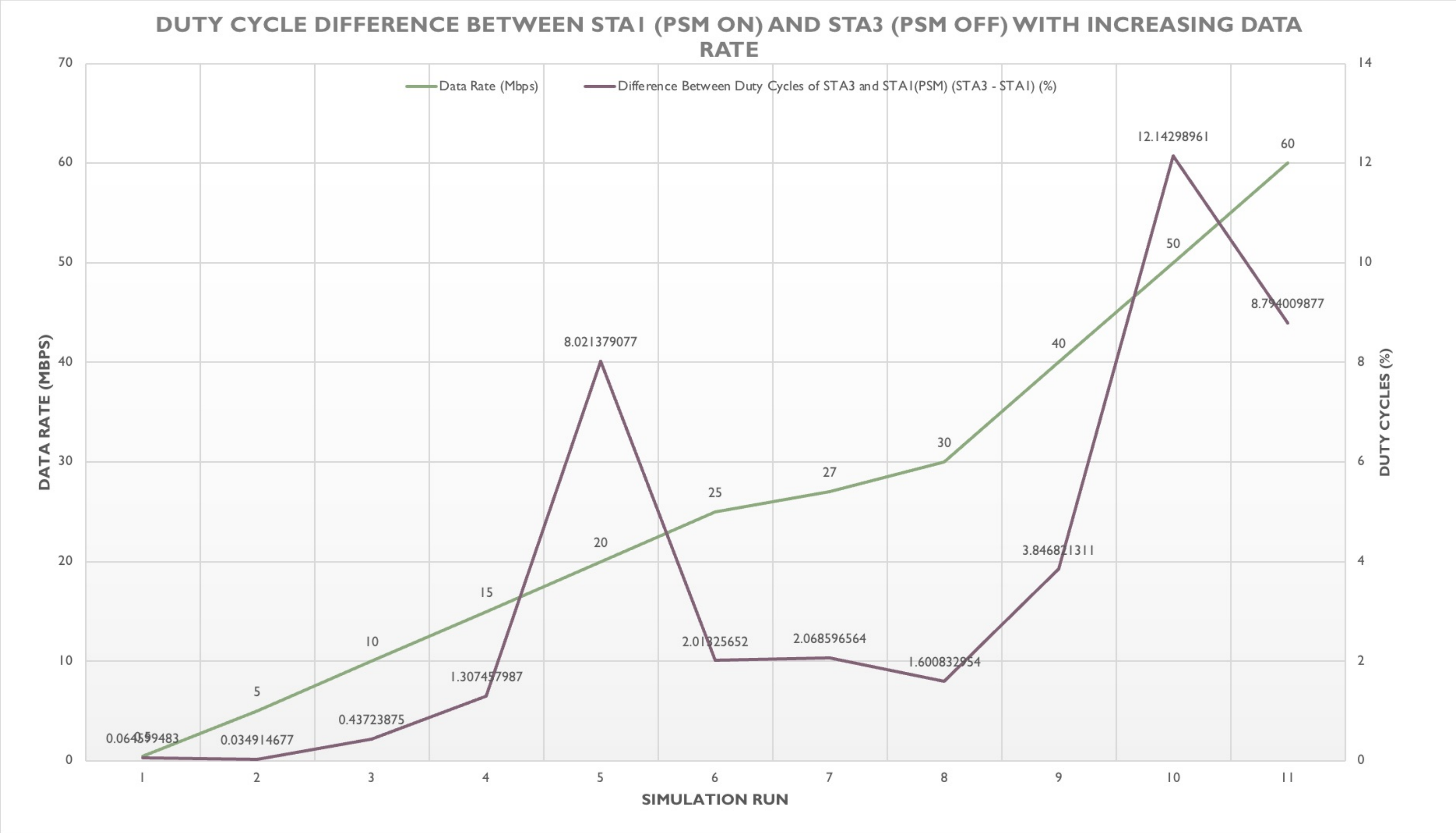
## RESULT 1

| DUTY CYCLE DIFFERENCE BETWEEN STA1 (PSM ON) AND STA2 (PSM OFF) WITH INCREASING DATA RATE | | |
|---|---|---|
| **Simulation No.** | **Data Rate (Mbps)** | **Difference Between Duty Cycles of STA2 and STA1(PSM) (STA2 - STA1) (%)** |
| 1 | 0.5 | 0 |
| 2 | 5 | 0.039279012 |
| 3 | 10 | 0.57015933 |
| 4 | 15 | 1.156860732 |
| 5 | 20 | 7.351698496 |
| 6 | 25 | 2.725639596 |
| 7 | 27 | 2.098317779 |
| 8 | 30 | 0.91429687 |
| 9 | 40 | 2.354370336 |
| 10 | 50 | 12.69009644 |
| 11 | 60 | 9.526844034 |

As we can see there is a clear difference between the duty cycles of stations 2 & 1. with increasing data rate it becomes more evident hitting a peak difference of 12.69%



| RESULT 2 | | |
|---|---|---|
| DUTY CYCLE DIFFERENCE BETWEEN STA1 (PSM ON) AND STA3 (PSM OFF) WITH INCREASING DATA RATE | | |
| **Simulation No.** | **Data Rate (Mbps)** | **Difference Between Duty Cycles of STA3 and STA1(PSM) (STA3 - STA1) (%)** |
| 1 | 0.5 | 0.064599483 |
| 2 | 5 | 0.034914677 |
| 3 | 10 | 0.43723875 |
| 4 | 15 | 1.307457987 |
| 5 | 20 | 8.021379077 |
| 6 | 25 | 2.01325652 |
| 7 | 27 | 2.068596564 |
| 8 | 30 | 1.600832954 |
| 9 | 40 | 3.846821311 |
| 10 | 50 | 12.14298961 |
| 11 | 60 | 8.794009877 |

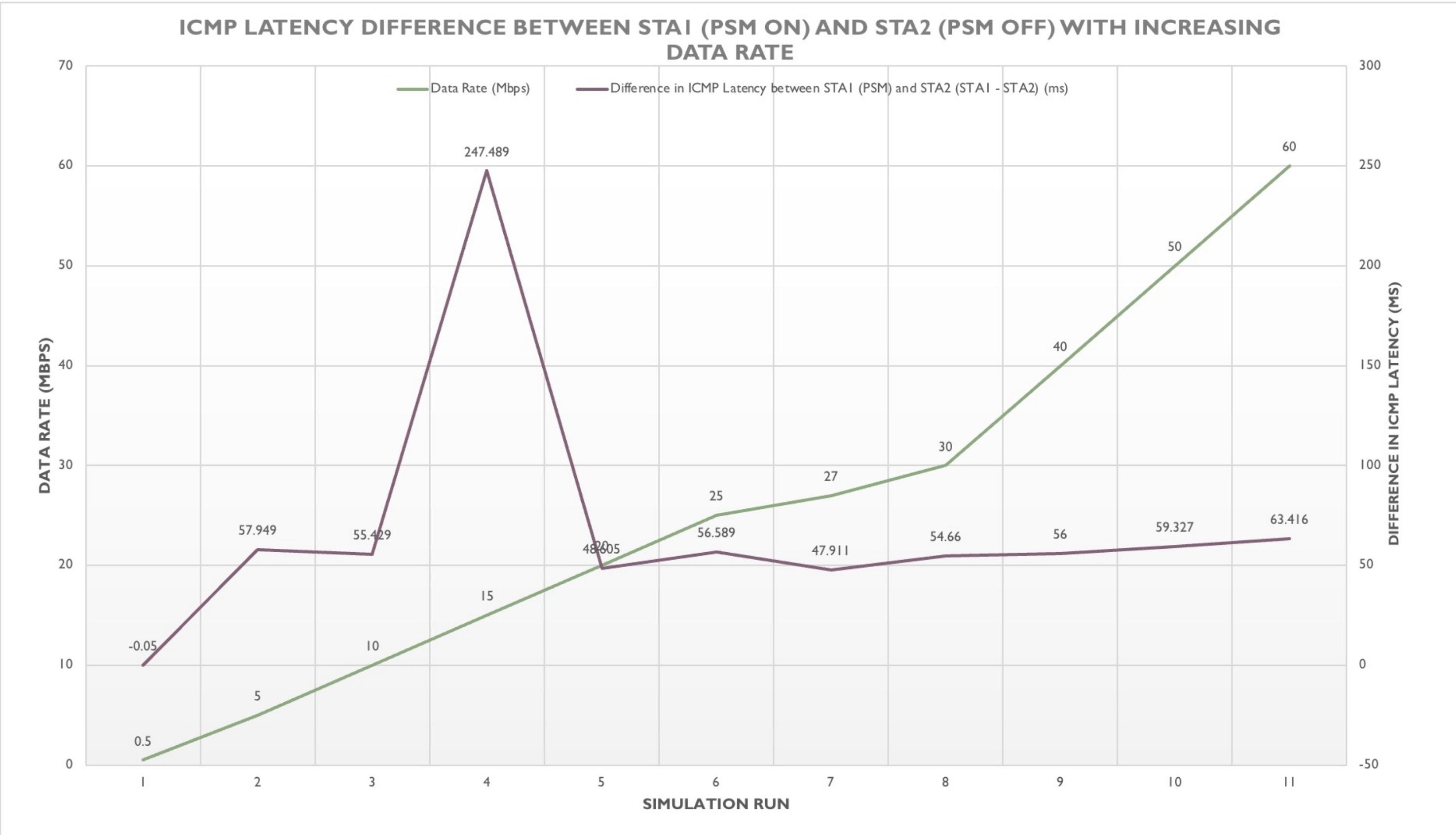



Again similar results can be seen when comparing stations 3 and 1. Station 1 performs worse due to Power Saving mode. hitting a similar peak of 12.14 %

## RESULT 3

ICMP LATENCY DIFFERENCE BETWEEN STA1 (PSM ON) AND STA2 (PSM OFF) WITH INCREASING DATA RATE

| Simulation No. | Data Rate (Mbps) | Difference in ICMP Latency between STA1 (PSM) and STA2 (STA1 - STA2) (ms) |
|---|---|---|
| 1 | 0.5 | -0.05 |
| 2 | 5 | 57.949 |
| 3 | 10 | 55.429 |
| 4 | 15 | 247.489 |
| 5 | 20 | 48.605 |
| 6 | 25 | 56.589 |
| 7 | 27 | 47.911 |
| 8 | 30 | 54.66 |
| 9 | 40 | 56 |
| 10 | 50 | 59.327 |
| 11 | 60 | 63.416 |

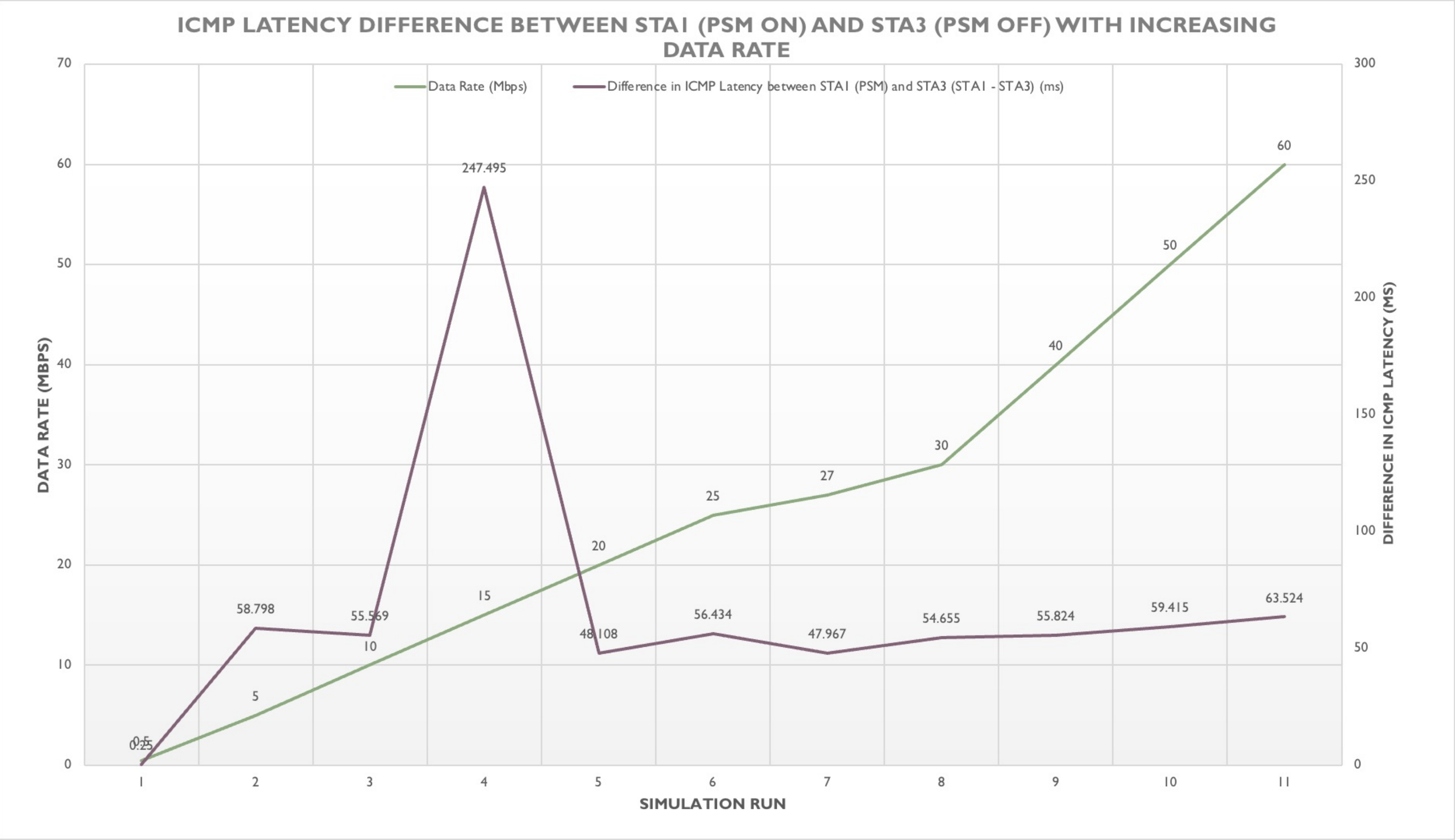



The difference in ICMP latency between stations 1 and 2 revolves around 45 to 60 ms. with the PSM enabled station 1 performing worse.

## RESULT 4

ICMP LATENCY DIFFERENCE BETWEEN STA1 (PSM ON) AND STA3 (PSM OFF) WITH INCREASING DATA RATE

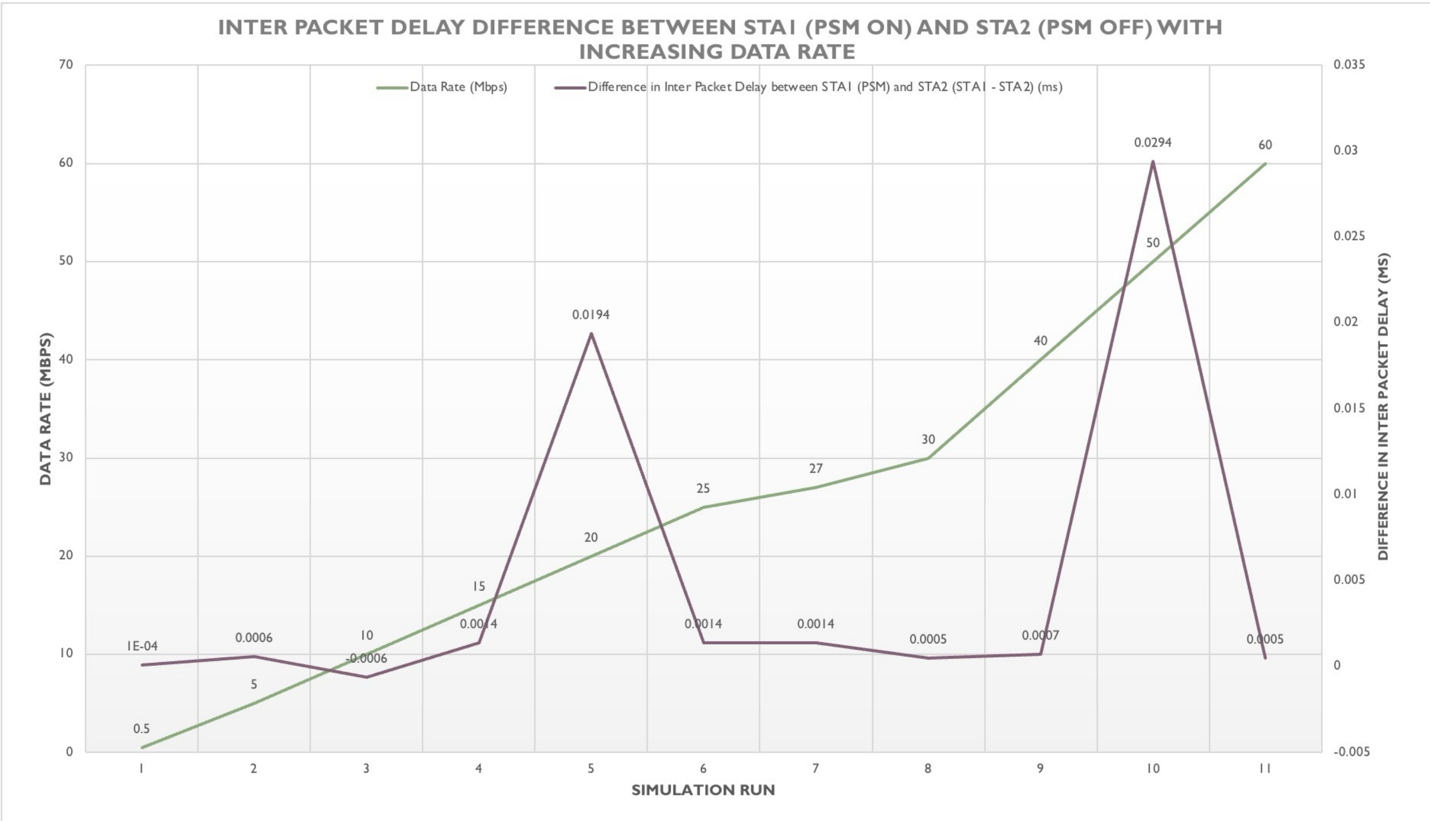
| Simulation No. | Data Rate (Mbps) | Difference in ICMP Latency between STA1 (PSM) and STA3 (STA1 - STA3) (ms) |
|---|---|---|
| 1 | 0.5 | 0.25 |
| 2 | 5 | 58.798 |
| 3 | 10 | 55.569 |
| 4 | 15 | 247.495 |
| 5 | 20 | 48.108 |
| 6 | 25 | 56.434 |
| 7 | 27 | 47.967 |
| 8 | 30 | 54.655 |
| 9 | 40 | 55.824 |
| 10 | 50 | 59.415 |
| 11 | 60 | 63.524 |




Similar results can be seen when comparing ICMP latency between stations 1 and 3.

Averaging around 50 to 65 ms.

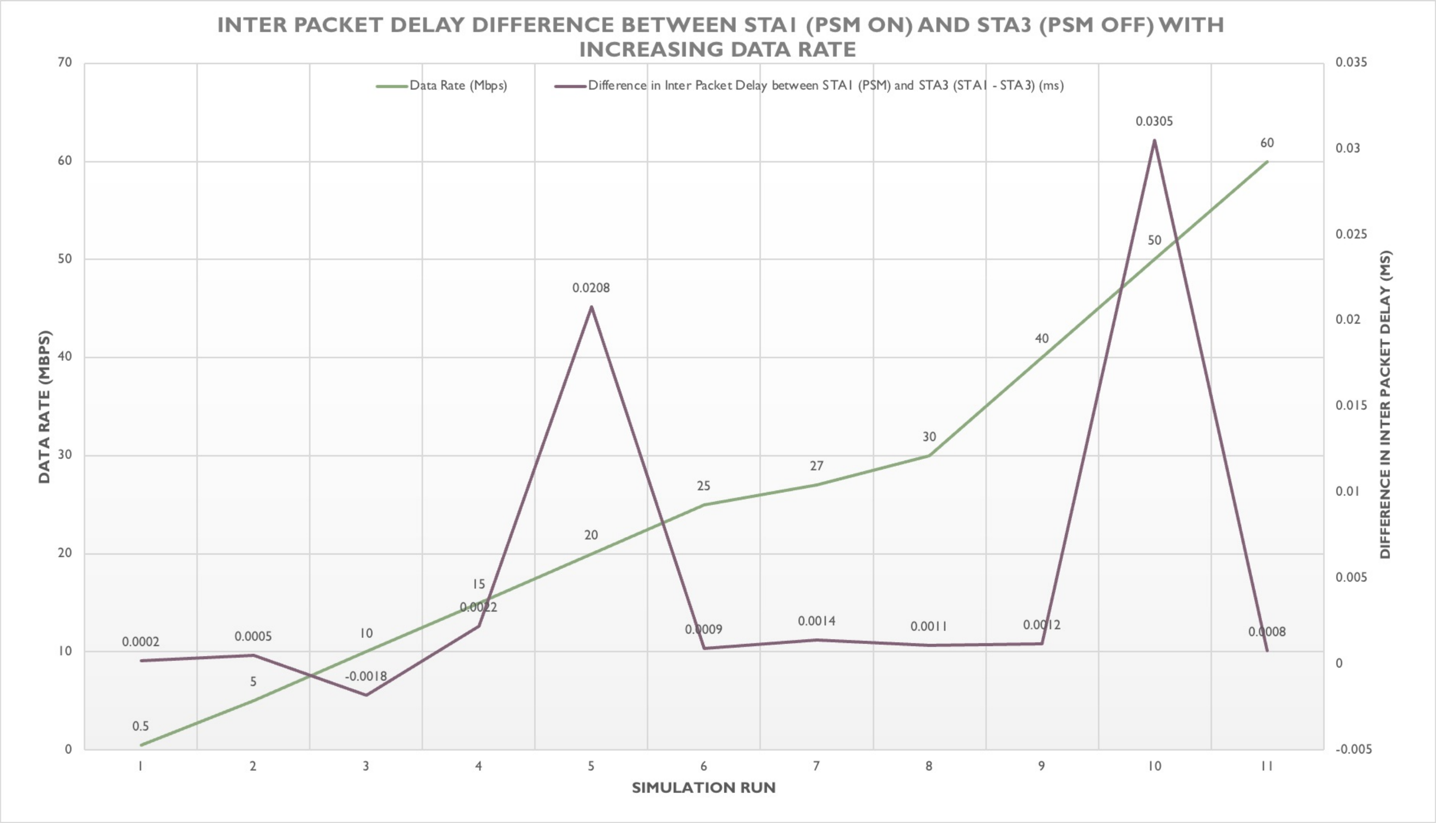
| RESULT 5 | | |
|---|---|---|
| INTER PACKET DELAY DIFFERENCE BETWEEN STA1 (PSM ON) AND STA2 (PSM OFF) WITH INCREASING DATA RATE | | |
| **Simulation No.** | **Data Rate (Mbps)** | **Difference in Inter Packet Delay between STA1 (PSM) and STA2 (STA1 - STA2) (ms)** |
| 1 | 0.5 | 1E-04 |
| 2 | 5 | 0.0006 |
| 3 | 10 | -0.0006 |
| 4 | 15 | 0.0014 |
| 5 | 20 | 0.0194 |
| 6 | 25 | 0.0014 |
| 7 | 27 | 0.0014 |
| 8 | 30 | 0.0005 |
| 9 | 40 | 0.0007 |
| 10 | 50 | 0.0294 |
| 11 | 60 | 0.0005 |




Inter packet latency of UDP packets reveal some differences with the PSM enabled station 1 performing worse than both stations 2 and 3

| RESULT 6 | | |
|---|---|---|
| INTER PACKET DELAY DIFFERENCE BETWEEN STA1 (PSM ON) AND STA3 (PSM OFF) WITH INCREASING DATA RATE | | |
| **Simulation No.** | **Data Rate (Mbps)** | **Difference in Inter Packet Delay between STA1 (PSM) and STA3 (STA1 - STA3) (ms)** |
| 1 | 0.5 | 0.0002 |
| 2 | 5 | 0.0005 |
| 3 | 10 | -0.0018 |
| 4 | 15 | 0.0022 |
| 5 | 20 | 0.0208 |
| 6 | 25 | 0.0009 |
| 7 | 27 | 0.0014 |
| 8 | 30 | 0.0011 |
| 9 | 40 | 0.0012 |
| 10 | 50 | 0.0305 |
| 11 | 60 | 0.0008 |

Similar results seen here too.



## Observations and Inferences on Results

i. **Result 1: Duty Cycle Difference between STA1 and STA2**

   A. **Observation**: The duty cycle difference between STA1 (with PSM) and STA2 (without PSM) fluctuates with the data rate. At the lowest data rates fluctuations are almost none but it spikes significantly after. There are huge spikes at 20Mbps, 50Mbps, and 60Mbps.

   B. **Inference**: PSM does not impact duty cycle as much at low data rate levels because it doesn't have to factor in as many power-saving adjustments. However, when more date rate is needed then PSM requires greater power-saving adjustments, it impacts the duty cycle a great deal. The spikes at the higher rates suggest that PSM works better or presents a greater adjustment difficulty with higher data transmission rates.

ii. **Result 2: Duty Cycle Difference between STA1 and STA3**

   A. **Observation**: Similar to Result 1, STA3's duty cycle divergence from STA1 is marginally less at low data rates but marginally more at higher data rates. But in contrast to STA2, the disparities at particular data rates (such 20 Mbps and 40 Mbps) are marginally greater.

   B. **Inference**: The effect of PSM on STA1 is consistent across comparisons with both STA2 and STA3. But variations in STA3's duty cycle values point to very slight variances in the way each non-PSM station communicates with STA1 at particular data speeds. This results from variations in transmission intervals or packet handling.

iii. **Result 3: ICMP Latency Difference between STA1 and STA2**

   A. **Observation**: ICMP latency discrepancies between STA1 and STA2 rises with larger data rates, averaging slightly about 50–60 ms at speeds higher than 25 Mbps.

   B. **Inference**: Power Saving Mode on STA1 results in huge delays in ICMP responses as the data rate increases slowly in our simulation. Also the ICMP responses during high throughput times are delayed because it takes longer to go active mode, power save mode, and then back to active mode again.

iv. **Result 4: ICMP Latency Difference between STA1 and STA3**

   A. **Observation**: The ICMP latency variance from STA1 to STA3 increases over time with increasing data rates and leveling off in and around 50ms to 60 ms latency just above 25 Mbps.

B. **Inference**: ICMP latency impacts from PSM are consistent regardless of whether `STA1` is compared with `STA2` or `STA3`. The delay is significant at higher data rates proving that PSM affects ICMP responsiveness as data demands increase. Yet this increase could be problematic for sensitive, time-critical applications.

v. **Result 5: Inter-Packet Delay Difference between STA1 and STA2**

A. **Observation**: Inter-packet delay differences between `STA1` and `STA2` are generally small. The fluctuations at lower data rates are lesser. Higher data rates results in minor variations showing slight increases of delay around 20 and 50 Mbps.

B. **Inference**: PSM on `STA1` at lower data rates has very little effect on inter-packet latency. This happens because a fewer number of packets need power management modifications. Though the impact is still minimal there are subtle fluctuations in delay as data rate rises. This in turn suggest that PSM may result in slight timing anomalies in packet treatment.

vi. **Result 6: Inter-Packet Delay Difference between STA1 and STA3**

A. **Observation**: Similar to Result 5, the inter-packet delay differences between `STA1` and `STA3` remain minimal at lower rates but exhibit minor variations at higher data rates. And the highest difference is noted to be around 50 Mbps.

B. **Inference**: At lower data rates, Power Saving Mode appears to have no effect on timing between UDP packets, based on the small inter-packet delay variances. Even though the effect is not major enough to result in significant timing problems. But the tiny increases in delay at higher rates suggests that PSM experiences slight delays in packet processing as a result of periodic switching.

## Summary of Findings

i. **Impact of PSM on Duty Cycle**: PSM has a greater effect on duty cycle differences at higher data rates. This is due to increased data demands necessitating more power management adjustments.

ii. **ICMP Latency**: PSM introduces significant latency for ICMP responses at higher data rates which for sure will impact latency-sensitive applications.

iii. **Inter-Packet Delay**: The impact of PSM on inter-packet delay is minimal, with only minor variations observed at higher data rates. This suggests that PSM's impact on UDP timing is limited.

These results collectively indicate that PSM is effective in conserving power but it can introduce latency at higher data rates, particularly for ICMP packets. In conclusion, we should keep in mind the trade-off between power saving and latency especially when using PSM in latency-sensitive scenarios.

3. Part 3 : Simulating Handoff Process Between Two Access Points

## Network Topology and Configuration

(a) **Topology Setup**

  i. We set up a simple network topology with one controller, two access points, and two stations.

  ii. **Network Components**

   A. **Stations**

   - `sta1`: Represents a station node with MAC address `00:00:00:00:00:02` and IP address `10.0.0.2/8`. Its initial position is set to (10, 40, 0).
   - `UDPS`: Represents a UDP server station with MAC address `00:00:00:00:00:04` and IP address `10.0.0.5/8`, positioned at (60, 20, 0).

   B. **Access Points (APs)**

   - `ap1`: The first access point in the network, with MAC address `00:00:00:00:10:02`, configured with SSID `handover`, operating mode `g`, and channel `1`. Its position is (15, 30, 0).
   - `ap2`: The second access point in the network, with MAC address `00:00:00:00:10:` also configured with SSID `handover`, mode `g`, and channel `6`. Its position is (55, 30, 0).

   C. **Controller**

   - `c1`: A controller node to manage the network components.

(b) **Propagation Model**

  i. We use a **log-distance propagation model** with an exponent parameter which is set to `5`. This affects signal strength and coverage behavior in the network.

(c) **Station Mobility and Handover**

  i. `sta1` is set to move across the network to simulate a handover between the two access points (`ap1` and `ap2`).

  ii. **Mobility Path**

   A. Initial position: (10, 40, 0)

   B. Start moving at time `2` seconds.

   C. Final position: (60, 40, 0), reaching at `25` seconds.

iii. **Mobility Control**

   A. Mobility is managed by the `net.startMobility`, `net.mobility`, and `net.stopMobility` commands which simulates the movement of `sta1` across the network.

(d) **Packet Capture**

   i. Packet capture is enabled on the interface `hwsim0`, which is used for capturing wireless traffic.

   ii. **Capture Configuration**

      A. We execute the command `tcpdump -i hwsim0 -w filename.pcap &` to start capturing packets on the `hwsim0` interface.

      B. Captured packets are saved in a `.pcap` file for later analysis.

(e) **Traffic Generation**

   i. **UDP Traffic Setup**

      A. The simulation includes a UDP server-client setup using `iperf3` to generate traffic between `sta1` (client) and `UDPS` (server).

      B. **Server** (`UDPS`): Runs an `iperf3` server listening on port `5566` with the command `iperf3 -s -p 5566 --logfile filename;`

      C. **Client** (`sta1`): Runs an `iperf3` client sending UDP traffic to `UDPS` on port `5566`, with a duration of `25` seconds, varying bandwidth between `0.2 Mbps to 3 Mbps` through a total of 11 simulations, using the command `iperf3 -c 10.0.0.5 -u -t 25 -b datarate -p 5566;`

(f) **Node Count and Configuration**

   i. **Total Nodes**: 5 nodes

      A. Stations: 2 (`sta1` and `UDPS`)

      B. Access Points: 2 (`ap1` and `ap2`)

      C. Controller: 1 (`c1`)

   ii. **MAC and IP Addresses**

      A. Each node has a unique MAC and IP address, as specified in the setup code.

(g) **WLAN Medium**

   i. **Medium Type**: `wmediumd` is used as the medium manager for the wireless network. Interference handling is enabled through the `interference` mode.

## Summary of Configuration and Procedures

(a) **Network Structure**

i. The network consists of two stations, two access points, and one controller.

(b) **Station Mobility**

i. `sta1` moves from `ap1` to `ap2`, simulating a handover.

(c) **Propagation Model**

i. A log-distance model with an exponent of `5` to simulate worst case signal degradation.

(d) **Packet Capture**

i. Packets are captured on `hwsim0` and stored in `sim11.pcap` for analysis.

(e) **UDP Traffic Test**

i. `iperf3` is used to generate UDP traffic between `sta1` and `UDPS` to study performance metrics.

These configurations will allow us to analyze handover behavior, packet delay, and the impact of mobility on wireless communication quality in a controlled, simulated environment.

## Enhancements to Handoff Simulation

In order to improve the handoff simulation, the following changes were implemented by us in order to reduce handoff times, packet loss, and reconnection time during mobility:

(a) **Background Scanning Configuration for Station (`sta1`)**

i. The `bgscan_module` parameter was set to `simple` to enable background scanning on `sta1`.

ii. `bgscan_threshold` was set to `-30 dBm`, which triggers a scan when signal strength drops below this threshold. This allows for faster detection of nearby access points with better signal strength.

iii. `s_interval` (short interval) was set to `2 seconds` and `l_interval` (long interval) to `10 seconds`. This ensures frequent scanning to discover and connect to stronger access points as the station moves.

(b) **Pre-association with Access Points**

i. Pre-association was configured for `sta1` with both `ap1` and `ap2`. This allows `sta1` to maintain connection credentials for multiple access points simultaneously thus enabling faster handoff by reducing the time needed for authentication and association during transitions.

(c) **User-Space Datapath for Access Points**

i. The `datapath` attribute for both `ap1` and `ap2` is set to `user` mode. This configuration allows for better control and flexibility of the handoff mechanism in a simulated environment. This can potentially reduce the reconnection time during access point transitions.

(d) **Enhanced Mobility Path and Timing**

i. The mobility path for `sta1` was defined with a starting position at `(15,20,0)` and ending position at `(65,20,0)`.

ii. Mobility timing was optimized with:

A. `sta1` starts moving at `1` second.

B. The station stops at `25` seconds, with `net.stopMobility` called at `27` seconds, providing a smooth handoff and reducing packet loss during the transition between access points.

These improvements will allow the mininet wifi simulation to perform faster handoffs with reduced packet loss thus achieving a more efficient and reliable connectivity experience for mobile stations.

## Result Generation

### Default Handoff Techniques

The default handoff calculation follows the steps below:

(a) **Step 1: Identify Deauthentication and Association Response Events**

We first extract the deauthentication event and the association response event from the captured packets.

i. **Deauthentication Event:** Occurs when the station (STA) sends a deauthentication frame to the original AP (AP1).

ii. **Association Response Event:** Occurs when the new AP (AP2) sends an association response to the STA.

The time difference between these two events is used to calculate the handoff delay:

$$\text{Handoff Delay} = \text{Association Response Time} - \text{Deauthentication Time}$$

(b) **Step 2: Identify UDP Packets for Reconnection Time**
Next, the UDP packets sent from the STA to a specific UDPS are analyzed before the deauthentication and after the association response.

i. The last UDP packet before deauthentication is recorded.
ii. The first UDP packet after the association response is recorded.

The time difference between these two UDP packets is used to calculate the reconnection time:

Reconnection Time = `First UDP Packet After Association Response - Last UDP Packet Before Deauthentication`

(c) **Step 3: Code Execution**
The Python code using `Scapy` processes the packets and calculates the above metrics using Scapy's packet manipulation functions. The key operations include:

i. Reading the PCAP file using `rdpcap`.
ii. Extracting deauthentication and association response frames.
iii. Extracting UDP packets from the STA to the UDPS.

**Improved Handoff Techniques**

The improved handoff calculation approach follows the steps described below:

(a) **Step 1: Find Authentication Request and Reassociation Response Events**
The first step involves identifying the authentication request sent by the STA to the new AP (AP2) and the reassociation response sent by AP2 to the STA.

i. **Authentication Request:** Sent from STA to AP2 during the handoff process.
ii. **Reassociation Response:** Sent from AP2 to STA after the authentication request.

The time difference between these two events is used to calculate the handoff delay:

`Handoff Delay = Reassociation Response Time - Authentication Request Time`

(b) **Step 2: Find Data Frames for Reconnection Time**

The second step involves analyzing the data frames exchanged between the STA and UDPS before the authentication request and after the reassociation response.

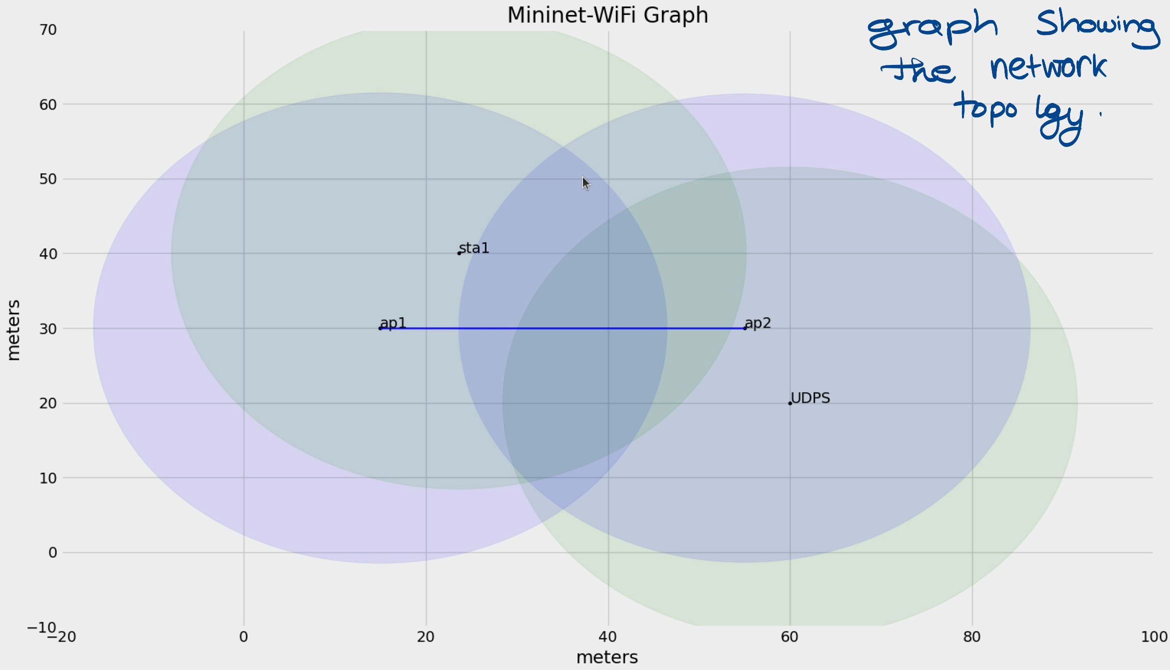
i. The last data frame before the authentication request is recorded.
ii. The first data frame after the reassociation response is recorded.

The time difference between these two data frames is used to calculate the reconnection time:

`Reconnection Time = First Data Frame After Reassociation Response - Last Data Frame Before Authentication Request`

(c) **Step 3: Code Execution** The Python code processes the packets and calculates the above metrics using Scapy's packet manipulation functions. The key operations include:

i. Reading the PCAP file using `rdpcap`.
ii. Extracting authentication request and reassociation response frames.
iii. Extracting data frames from the STA to the UDPS.



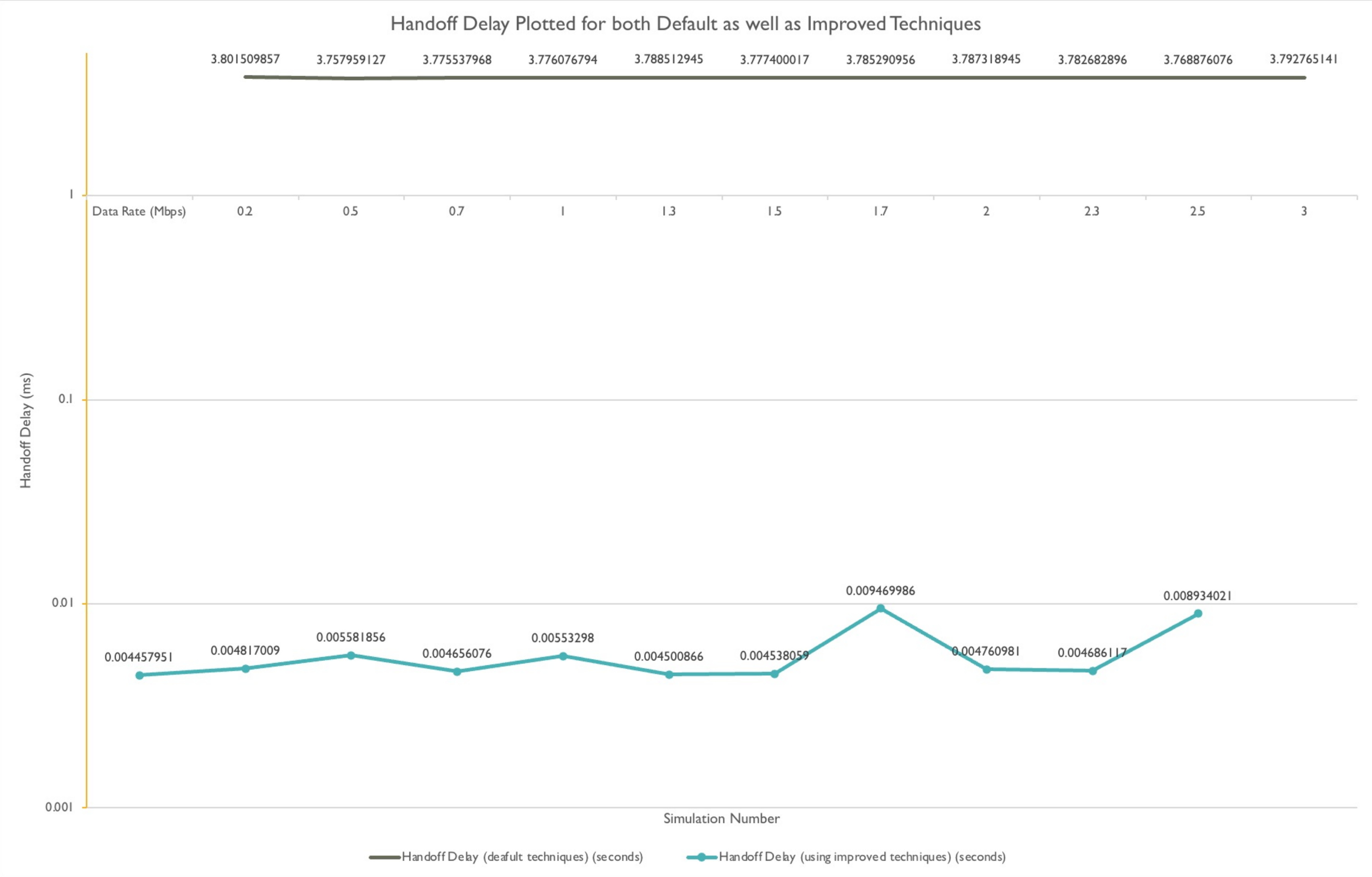

graph showing the network topo lgy.

# Results from Multiple Simulations

| HANDOFF ANALYSIS ( DEFAULT TECHNIQUES ) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Simulation No.** | **Data Rate (Mbps)** | **Time (seconds)** | **Transfer (MB) (at receiver)** | **Bitrate (Mbps) (at receiver)** | **Jitter (ms) (at receiver)** | **Total Packets Transferred during Handoff** | **Total Packets Lost during Handoff** | **Handoff Delay (seconds)** | **Reconnection Time (seconds)** |
| 1 | 0.2 | 25 | 0.532 | 0.174 | 0.094 | 92 | 56 | 3.801509857 | 4.157729149 |
| 2 | 0.5 | 25 | 1.34 | 0.449 | 0.106 | 230 | 110 | 3.757959127 | 4.110759974 |
| 3 | 0.7 | 25 | 1.9 | 0.636 | 0.086 | 323 | 135 | 3.775537968 | 4.118535995 |
| 4 | 1 | 25 | 2.83 | 0.948 | 0.115 | 459 | 110 | 3.776076794 | 4.101925135 |
| 5 | 1.3 | 25 | 3.71 | 1.24 | 0.117 | 597 | 122 | 3.788512945 | 4.095789909 |
| 6 | 1.5 | 25 | 4.29 | 1.44 | 0.097 | 690 | 130 | 3.777400017 | 4.110767126 |
| 7 | 1.7 | 25 | 4.78 | 1.6 | 0.046 | 779 | 211 | 3.785290956 | 4.098469019 |
| 8 | 2 | 25 | 5.7 | 1.91 | 0.063 | 719 | 187 | 3.787318945 | 4.106613874 |
| 9 | 2.3 | 25 | 6.57 | 2.2 | 0.077 | 922 | 203 | 3.782682896 | 4.084235907 |
| 10 | 2.5 | 25 | 7.15 | 2.4 | 0.086 | 628 | 215 | 3.768876076 | 4.106253862 |
| 11 | 3 | 25 | 8.58 | 2.87 | 0.131 | 830 | 260 | 3.792765141 | 4.083073854 |

| HANDOFF ANALYSIS ( IMPROVED TECHNIQUES) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Simulation No.** | **Data Rate (Mbps)** | **Time (seconds)** | **Transfer (MB) (at receiver)** | **Bitrate (Mbps) (at receiver)** | **Jitter (ms) (at receiver)** | **Total Packets Transferred during Handoff** | **Total Packets Lost during Handoff** | **Handoff Delay (seconds)** | **Reconnection Time (seconds)** |
| 1 | 0.2 | 25 | 0.607 | 0.198 | 20.982 | 36 | 3 | 0.004457951 | 1.093482971 |
| 2 | 0.5 | 25 | 1.45 | 0.485 | 10.118 | 70 | 32 | 0.004817009 | 1.076954126 |
| 3 | 0.7 | 25 | 2.02 | 0.677 | 9.659 | 122 | 47 | 0.005581856 | 1.033838987 |
| 4 | 1 | 25 | 2.93 | 0.981 | 6.667 | 113 | 38 | 0.004656076 | 1.057322979 |
| 5 | 1.3 | 25 | 3.74 | 1.25 | 6.846 | 203 | 100 | 0.00553298 | 1.032400131 |
| 6 | 1.5 | 25 | 4.3 | 1.44 | 1.894 | 233 | 124 | 0.004500866 | 1.048754215 |
| 7 | 1.7 | 25 | 4.84 | 1.62 | 4.91 | 265 | 135 | 0.004538059 | 1.032751083 |
| 8 | 2 | 25 | 5.61 | 1.88 | 10.686 | 213 | 163 | 0.009469986 | 1.032920837 |
| 9 | 2.3 | 25 | 6.62 | 2.22 | 4.433 | 347 | 40 | 0.004760981 | 1.046067953 |
| 10 | 2.5 | 25 | 7.03 | 2.35 | 3.162 | 433 | 208 | 0.004686117 | 1.038084984 |
| 11 | 3 | 25 | 8.26 | 2.76 | 3.906 | 520 | 271 | 0.008934021 | 1.061205864 |

| RESULT 1 | | | |
|---|---|---|---|
| Handoff Delay Plotted for both Default as well as Improved Techniques | | | |
| Simulation No. | Data Rate (Mbps) | Handoff Delay (deafult techniques) (seconds) | Handoff Delay (using improved techniques) (seconds) |
| 1 | 0.2 | 3.801509857 | 0.004457951 |
| 2 | 0.5 | 3.757959127 | 0.004817009 |
| 3 | 0.7 | 3.775537968 | 0.005581856 |
| 4 | 1 | 3.776076794 | 0.004656076 |
| 5 | 1.3 | 3.788512945 | 0.00553298 |
| 6 | 1.5 | 3.777400017 | 0.004500866 |
| 7 | 1.7 | 3.785290956 | 0.004538059 |
| 8 | 2 | 3.787318945 | 0.009469986 |
| 9 | 2.3 | 3.782682896 | 0.004760981 |
| 10 | 2.5 | 3.768876076 | 0.004686117 |
| 11 | 3 | 3.792765141 | 0.008934021 |

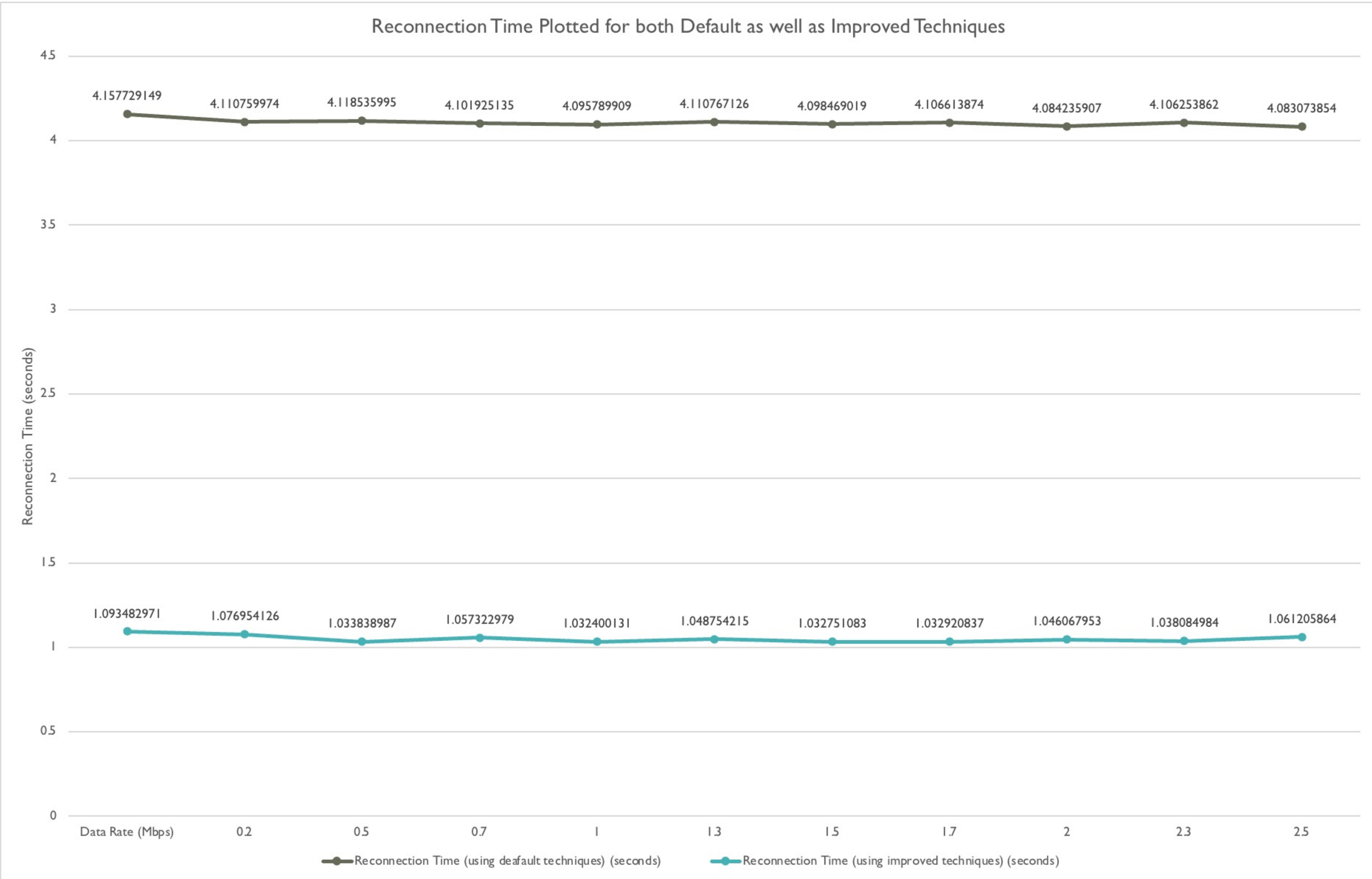



As we can see from the results on the left, Handoff delay with default techniques takes almost 3.8 seconds consistently even with varying data rates.
whereas handoff delay when improved using various techniques had values ranging from 0.004 seconds to 0.009 seconds.

## RESULT 2

Reconnection Time Plotted for both Default as well as Improved Techniques

| Simulation No. | Data Rate (Mbps) | Reconnection Time (using deafault techniques) | Reconnection Time (using improved techniques) (sec |
|---:|---:|---:|---:|
| 1 | 0.2 | 4.157729149 | 1.093482971 |
| 2 | 0.5 | 4.110759974 | 1.076954126 |
| 3 | 0.7 | 4.118535995 | 1.033838987 |
| 4 | 1 | 4.101925135 | 1.057322979 |
| 5 | 1.3 | 4.095789909 | 1.032400131 |
| 6 | 1.5 | 4.110767126 | 1.048754215 |
| 7 | 1.7 | 4.098469019 | 1.032751083 |
| 8 | 2 | 4.106613874 | 1.032920837 |
| 9 | 2.3 | 4.084235907 | 1.046067953 |
| 10 | 2.5 | 4.106253862 | 1.038084984 |
| 11 | 3 | 4.083073854 | 1.061205864 |




Reconnection Time is defined as the time difference between the last UDP packet Transfered before hand off and the first UDP packet transfered after handoff.

Default handoff techniques showcased around 4 seconds of reconnection time. which was higher by 3 seconds from the improved handoff Techniques. with 1 second of reconnection Time.

# RESULT 3

Packets Lost During Handoff Plotted for both Default as well as Improved Techniques

| Simulation No. | Data Rate (Mbps) | Total Packets Lost during Handoff (using defaul | Total Packets Lost during Handoff ( using improved t |
|---|---|---|---|
| 1 | 0.2 | 56 | 3 |
| 2 | 0.5 | 110 | 32 |
| 3 | 0.7 | 135 | 47 |
| 4 | 1 | 110 | 38 |
| 5 | 1.3 | 122 | 100 |
| 6 | 1.5 | 130 | 124 |
| 7 | 1.7 | 211 | 135 |
| 8 | 2 | 187 | 163 |
| 9 | 2.3 | 203 | 40 |
| 10 | 2.5 | 215 | 208 |
| 11 | 3 | 260 | 271 |


Across both default and improved handoff techniques. we can see that with increasing data rate the no. of packets lost during handoff also increases. Also the packets lost during handoff are significantly lessor in the improved handoff techniques.

## Observations and Inference on Results

Based on the results from the handoff simulation, the following inferences and observations are made by us for each of the metrics:

(a) **Result 1: Handoff Delay**

i. **Observation:** The handoff delay when using default techniques remains relatively high. The values are consistently around 3.75 to 3.80 seconds across various data rates. In contrast, the improved techniques reduce the handoff delay dramatically to values ranging from approximately 0.0045 to 0.009 seconds.

ii. **Inference:** The use of improved handoff techniques, such as background scanning, pre-association, and optimized mobility paths, has led to a significant reduction in handoff delay. This improvement is especially critical in applications requiring minimal service interruption. The improved techniques are highly effective in reducing handoff delay by maintaining better awareness of nearby access points.

(b) **Result 2: Reconnection Time**

i. **Observation:** The reconnection time with default techniques is consistently around 4.08 to 4.15 seconds, while with improved techniques, it drops significantly to around 1.03 to 1.09 seconds.

ii. **Inference:** This reduction indicates that the improvements also positively impact the time required for the station to reconnect to the network after a handoff. These optimizations are due to background scanning, pre-association with access points and the user-space datapath configuration. The optimizations streamline the reconnection process, reducing the time for re-establishing the connection.

(c) **Result 3: Packet Loss During Handoff**

i. **Observation:** The packet loss during handoff with default techniques varies considerably. The packet losses are increasing in nature as the data rate increases. In default settings, packet loss ranges from 56 packets at a 0.2 Mbps data rate to 260 packets at a 3 Mbps rate. With the improved techniques, packet loss is drastically reduced in most cases. These can be seen especially at lower data rates (e.g., from 56 to 3 at 0.2 Mbps).

ii. **Inference:** The reduction in packet loss at lower to moderate data rates suggests that the improved techniques are highly effective in reducing packet loss during handoff. However, at higher data rates (3 Mbps) the packet loss remains significant even with improvements. This occurs due to the limits of available bandwidth or network congestion during high-throughput transmissions. But overall improved techniques showcase better results than the default techniques.

## Summary of Findings

For us the improved handoff techniques have successfully reduced both the handoff delay and reconnection time by a large margin, as well as minimized packet loss. These enhancements are beneficial for maintaining quality of service (QoS) in applications where low latency and minimal packet loss are critical.