

## Group 7: Ayala Wang, Shashank Bhagwani, Shiyuan Wang, Nandhan Natarajan

Link to GitHub Repo: <https://github.com/AyalaWang/cs190Nproject.git>

## Campus Network Speed Test Analysis - Experiment Plan (Checkpoint 1)

### 1. High-Level Explanation of the Experiment

- **Objective:** Analyze UCSB campus Wi-Fi performance across various locations and environmental conditions to identify factors impacting network quality, specifically download/upload speed, latency, jitter, and packet loss. This analysis aims to provide UCSB IT services with actionable insights to improve connectivity on campus.
- **Data Type:** The key metrics to collect include:
  - **Download/upload speed:** Measures data transfer rate between network and devices.
  - **Latency:** The delay between a data request and response.
  - **Jitter:** Variation in packet arrival time, affecting real-time services.
  - **Packet loss:** Percentage of data packets lost during transmission.
- **Metadata:** Each data point will include:
  - **Location:** Specific campus areas (e.g., libraries, lecture halls, outdoor plazas).
  - **Time:** Peak (high usage) and off-peak (low usage) times.
  - **Environmental conditions:** Indoor vs. outdoor setting, potentially impacting network signal and stability.

### 2. Pseudo Code for Data Collection Pipeline (in python)

```
# Pseudo code for Wi-Fi Performance Data Collection using netUnicorn

from netunicorn import Experiment, Task, Device

# Step 1: Define Experiment Setup
experiment = Experiment(name="UCSB Wi-Fi Performance Analysis")

# Step 2: Configure Device List and Tasks
# Devices are Raspberry Pis or mobile devices in specific campus locations
devices = [
    Device(id="library_pi", location="library"),
    Device(id="lecture_hall_pi", location="lecture_hall"),
    Device(id="outdoor_pi", location="outdoor_plaza")
]
```

```

# Step 3: Define netUnicorn Tasks
# Assuming netUnicorn has built-in tasks for network metrics collection
start_speed_test_task = Task("start_speed_test") # Initiates the test
collect_metrics_task = Task("collect_metrics", params=["download_speed", "upload_speed", "latency", "packet_loss"])
end_speed_test_task = Task("end_speed_test") # Completes the test

# Step 4: Assign Tasks to Each Device and Configure Metadata
for device in devices:
    experiment.add_task(start_speed_test_task, device)
    experiment.add_task(collect_metrics_task, device, metadata={
        "location": device.location,
        "time_period": "peak" if within_peak_hours() else "off-peak",
        "environmental_condition": "indoor" if device.location != "outdoor_plaza" else "outdoor"
    })
    experiment.add_task(end_speed_test_task, device)

# Step 5: Execute Experiment and Collect Data
experiment.run()
results = experiment.collect_results()
for result in results:
    store_data(result.data, storage="database")
    # Metadata will ensure that data is labeled by location, time, and condition.

# Step 6: Finalize and Clean Up
experiment.finalize()

```

### 3. Task References or Implementation

Each task in the pipeline is designed to capture the necessary network performance metrics effectively. Below is the description and implementation approach for each task:

- **Task 1: start\_speed\_test\_task**
  - **Description:** This task initiates the speed test on the device, setting up any required configurations for data collection.
  - **Implementation Reference:**
    - \* If netUnicorn has a prebuilt `start_speed_test_task`, it can be used directly. Otherwise, create a custom task using shell commands or tools like `iperf3` to begin the network test.
- **Task 2: collect\_metrics\_task**
  - **Description:** The main data collection task, which gathers download/upload speed, latency, jitter, and packet loss metrics. This task should run during peak and off-peak times and in varied environmental conditions to provide a comprehensive dataset.
  - **Implementation Reference:**
    - \* Use netUnicorn's `collect_metrics_task` if available, spec-

ifying parameters like "download\_speed", "upload\_speed", "latency", "jitter", and "packet\_loss". If a predefined task isn't available, a custom task can be implemented using network measurement tools to retrieve these metrics.

- **Task 3: end\_speed\_test\_task**
  - **Description:** This task terminates the speed test and ensures all metrics have been logged properly.
  - **Implementation Reference:**
    - \* Use netUnicorn's `end_speed_test_task` if it exists. If not, implement a simple cleanup function to finalize the test process, ensuring all data is saved.
- **Metadata Tagging**
  - **Description:** After collecting metrics, each data point will be tagged with metadata to include information on location, time (peak/off-peak), and environmental conditions.
  - **Implementation:** Implement metadata tagging directly within `collect_metrics_task`, using location, time, and condition parameters set for each device in the experiment.