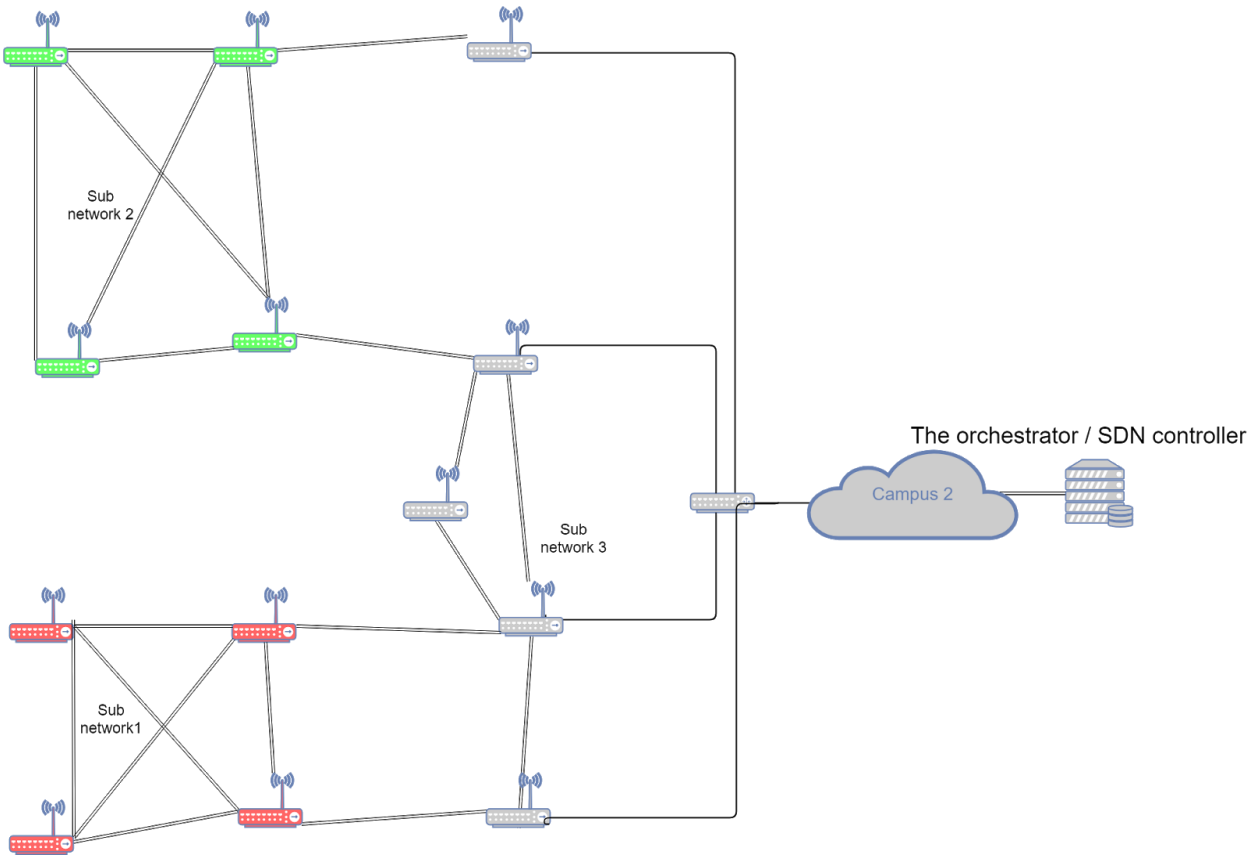
SDN based Mobile Ad-hoc NETwork (MANET) operating system

Abstract:

Mobile Ad-hoc NETworks (MANET) is a communication platform for first response units, characterized by its rapidly changing connectivity and bandwidth over the communication links. At the same time, an application runs on first response units often requires strict availability of end to end bandwidth and delay. For example, if a certain link bandwidth is 1.5Mbps and user video stream requires 1Mbps, the best solution is to allow not more than one stream at a time, otherwise none of the users will get satisfactory Quality of Service (QoS).

In this project, we consider creating a centralized Software Defined Network (SDN) controller and an orchestrator that can guarantee network resources to an application according to its constraints and the current network topology. The orchestrator should receive application stream request and calculate the optimal route for it. In case the network does not have enough resources to serve the application, the request is rejected. The SDN controller task is to receive optimal computed path from the orchestrator and to install it over the relevant nodes. Our assumption is that the network has basic IP connectivity from the start (ping from any node to any node) and link metrics are known precisely at any time.



Goals:

- Learning Stage
  - SDN, OpenFlow, RYU Controller
  - Mininet Tutorials
  - NRL MGEN tool

- Create Mininet network topology database
  - Each item in database will be a JSON file that will describe the topology
  - The file will include:
    - Nodes (hosts and switchesMininet) that appear in the topology
    - Links between nodes with the following attributes:
      - Link bandwidth
      - Link delay (queuing delay can be ignored)

- Create application traffic scenario database
  - Each item in database will be a JSON file that will include batch of unicast streams of data from source to destination hosts
  - Each stream will have the following attributes:
    - Minimal bandwidth
    - Maximum delay
    - Priority (High and Low)

  - Implement SDN Application that supports 2 routing algorithm options: Shortest Path & Advanced
    - Shortest Path Routing algorithm
      - The algorithm will receive a network topology JSON file as input and will calculate shortest path routing between every two hosts
        - Link cost is defined as [1/bandwidth]

    - Advanced Routing algorithm
      - The algorithm will receive as input an application traffic scenario and a network topology as input and will calculate the path for each stream.
      - The calculation can be based on one of the following (TBD):
        - min-cost-flow problem solution
        - constrained shortest path problem solution
  - For each algorithm: the SDN Application will configure the SDN controller with the routing paths calculated by the algorithm. This will be done for each specific network scenario and application scenario.

  - Per each algorithm and topology – application traffic scenario combination: Calculate the *TrafficScore* - Using the MGEN tool

- Note: Control plane traffic must be taken into account in the case of advanced routing algorithm. The Control plane affect on Traffic performance is: x% of total network bandwidth to be added on each link (where x=0,5,10,15,20)

- The *TrafficScore* will be:

$$\sum_{priority} (\#Successful\ Streams\ \ of\ specific\ priority) * (Total\ \#\ of\ streams)\text{^}priority$$

  - *SuccessfulStream* - stream that did not experience congestion/excessive delay on its path. Calculated path bandwidth should be compared to the stream expected bandwidth. Calculated path delay should be compared to stream Maximum delay.

- Mininet Simulation special notes:
  - The SDN application will be aware in each simulation at the beginning about the network topology and the traffic scenario.
  - Topology will be persistent during single simulation.
  - Traffic flows will be persistent during each single simulation

Requirements:

Introduction to Networking (Must) , Internet Networking (Optional)

Programming Language:
Python

Guided by:
Evgeny Reznik, Dr. Hanit Giat, Prof. Reuven Cohen

Collaborate with: