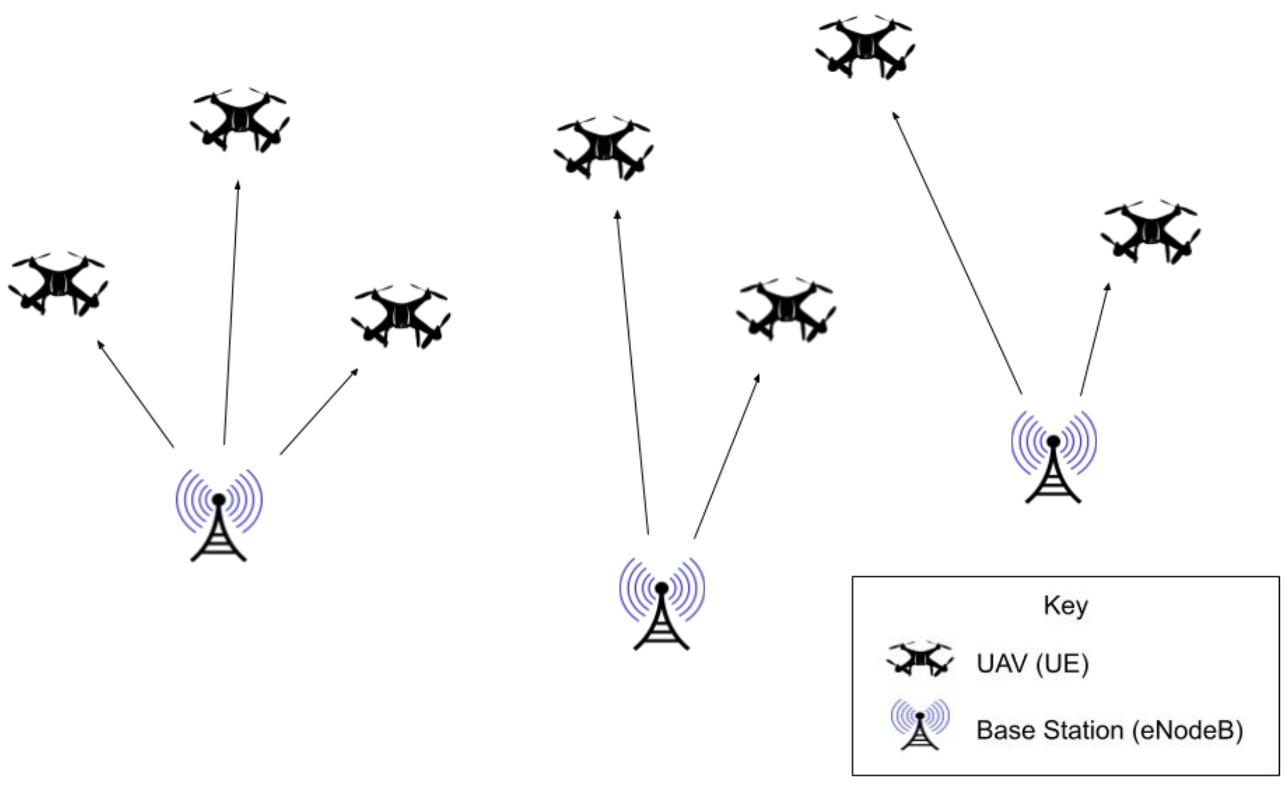
Open-source Prototyping of 5G Wireless Systems for Unmanned Aerial Vehicles
sdmay21-13

Students: Daniel Shaw, Ryan Ingram, Suraj Karn, Tyler Hutchinson
Faculty Adviser: Dr. Hongwei Zhang

Introduction

Problem: Current 4G & 4G LTE networks lack fast speeds, bandwidth, and low-latency required for ultra-reliable unmanned drone and future device communication

Solution: An integrated 5G network simulator and unmanned aerial vehicle (UAV) simulator would allow us to model connections in a 5G cellular network. The simulator would help test reliable connection beyond the line of sight, and enables replication of scenarios such as last-mile delivery and safety inspections.

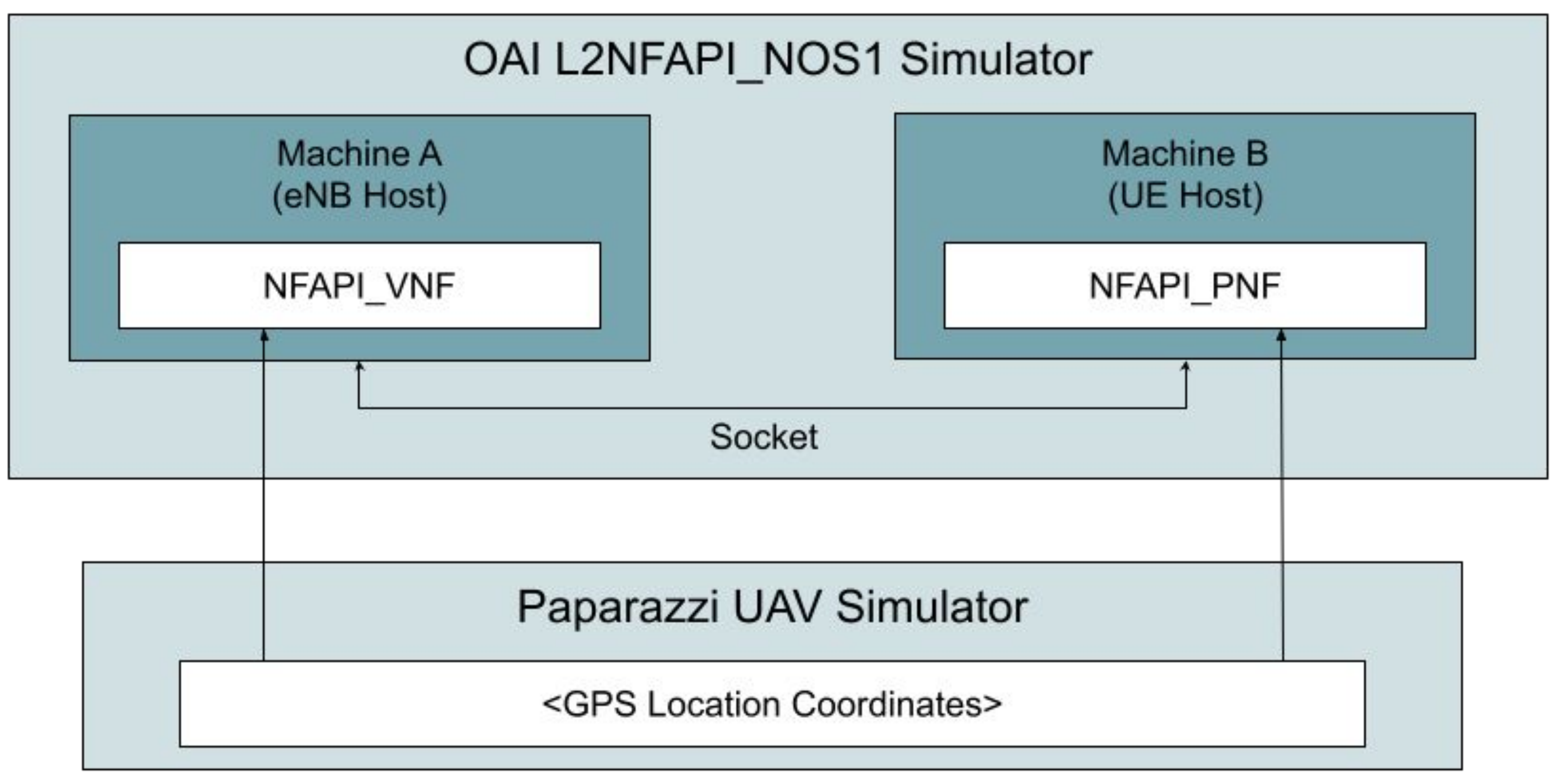


Intended Users and Uses

Academic Researchers: The project is aimed to be used in a research environment. The research would then help in expanding students and specialist related projects.

Industry Professionals: Could be used for multiple different industries to increase efficiency and productivity.

Industry Research: Project can be expanded upon by other professionals to increase the value of the software and project for more specialized industry roles.


OAI L2NFAPI_NOS1 Simulator

Machine A (eNB Host)

NFAPI_VNF

Machine B (UE Host)

NFAPI_PNF

Socket

Paparazzi UAV Simulator

<GPS Location Coordinates>

Design Requirements

Functional Requirements:
- Simulate a multi-cell environment
  - Multiple evolved node base stations (eNB) and multiple user equipments (UE)
- Supports simulation of different networking algorithms

Non-functional Requirements:
- Simulation must be easy to run
- UAV and network simulators must operate smoothly together

Engineering Constraints:
- Some network simulators require multiple machines (linux kernel policy will drop packets with local source addresses)
- OpenAirInterface (OAI) network simulators only support single-cell environments

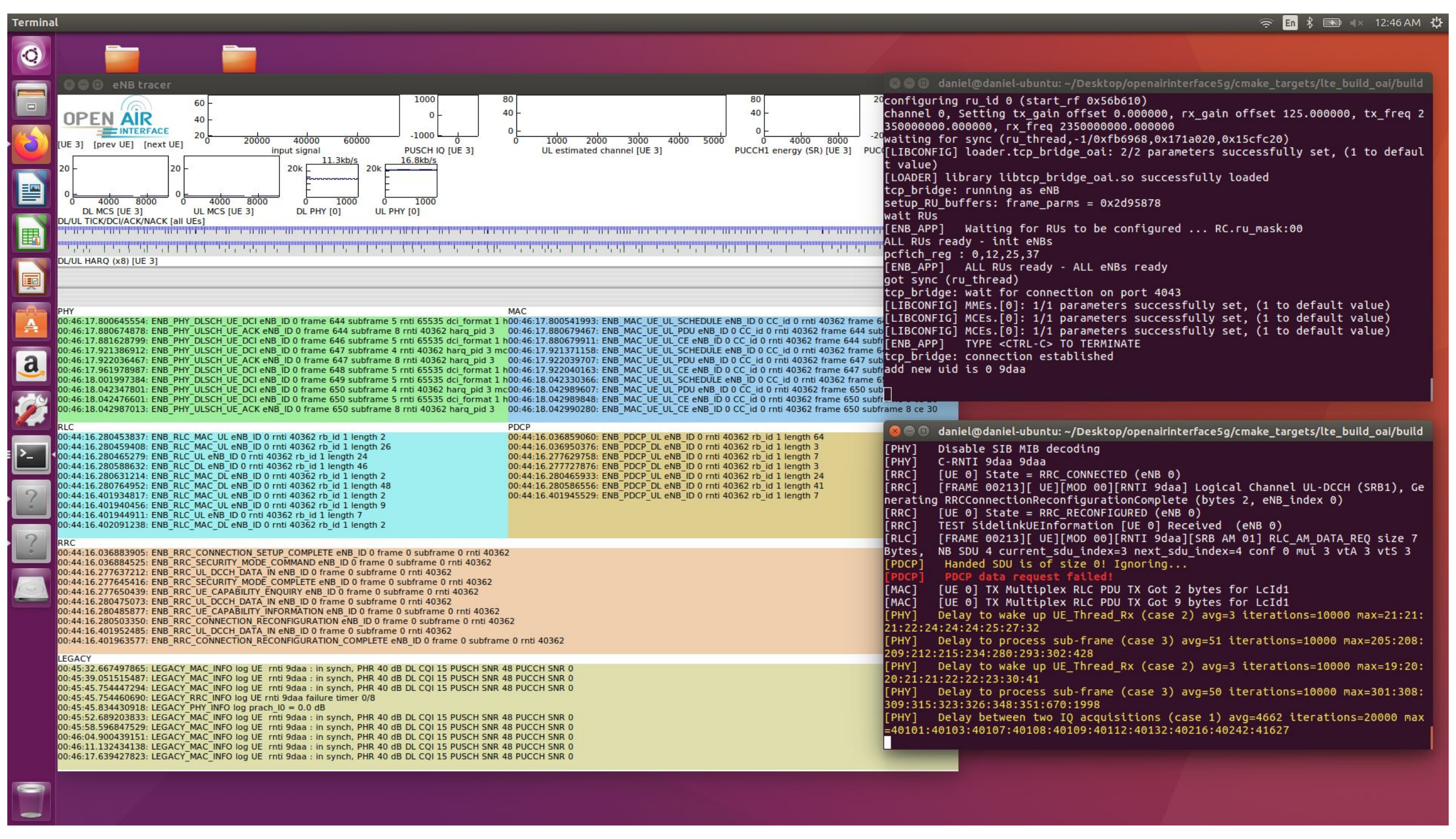
Operating Environment:
- Ubuntu 16.04 with low-latency kernel 4.4.0 or higher

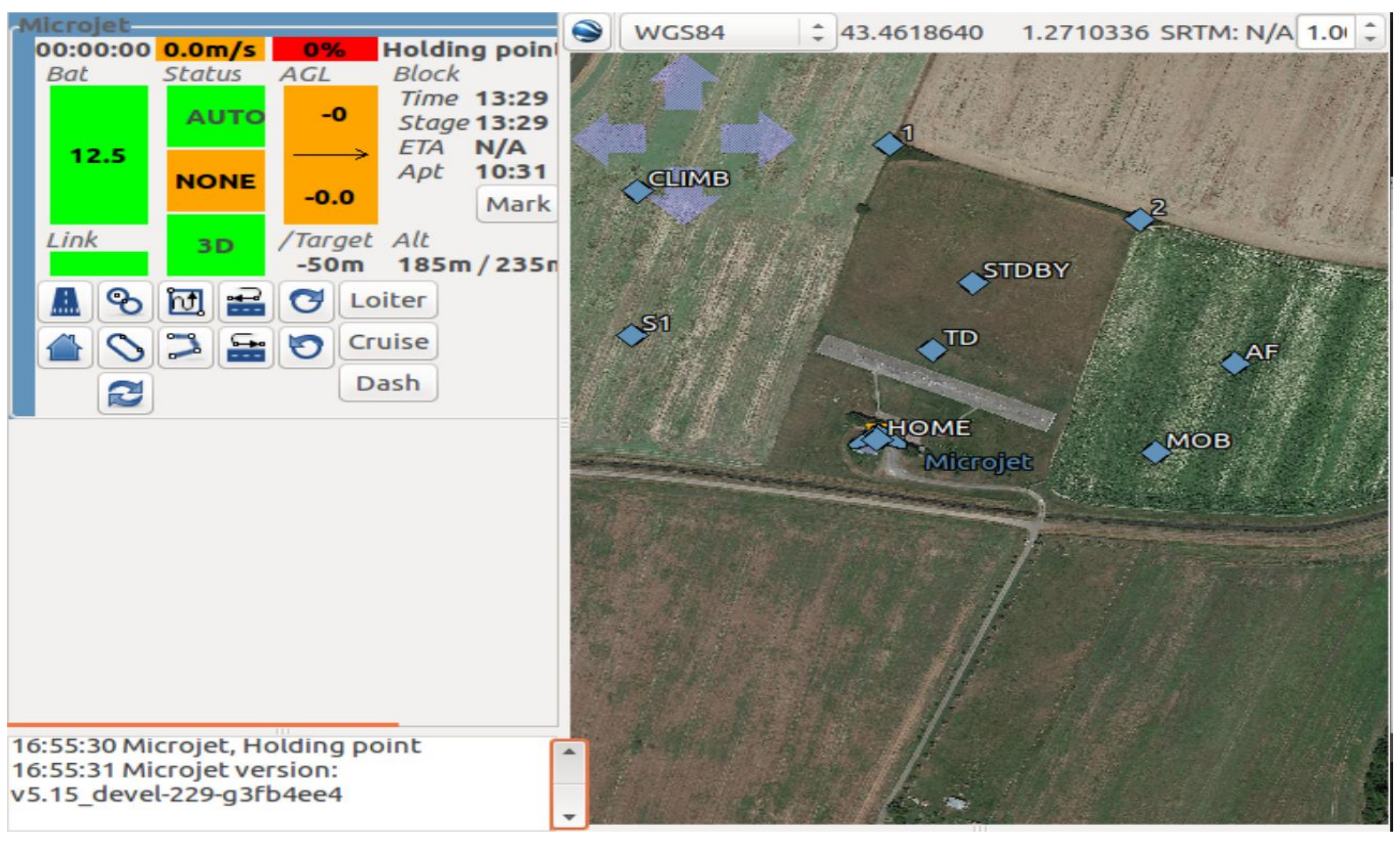
Relevant Standards:
- 3rd Generation Partnership Project (3GPP) Specifications for Cellular Telecommunication (Release 10, 12, and 14)
- Network Functional Platform Interface (nFAPI) Specifications by Small Cell Forum (Release 9)

Design Approach

Approach: Use OAI's L2NFAPI_NOS1 simulator as our network simulator. Then, retrieve GPS location coordinates of drones and base stations from Paparazzi UAV Simulator and inject them into where location coordinates are stored for the eNB and UE in OAI. This would allow us to co-simulate network connectivity, with OAI, during UAV flight patterns in Paparazzi.

Design Challenges/Shortcomings:
- Outdated or unclear OAI tutorials
- Strict hardware/software requirements needed to run OAI simulators
- Unable to get OAI simulators fully running




Microjet

Bat	Status	AGL	Block
00:00:00	0.0m/s	0%	Holding poin
12.5	AUTO	-0	Time 13:29
	NONE	-0.0	Stage 13:29
			ETA N/A
			Apt 10:31
Link	3D	/Target -50m	Alt 185m / 235n

WGS84 43.4618640 1.2710336 SRTM: N/A 1.0

```
16:55:30 Microjet, Holding point
16:55:31 Microjet version:
v5.15_devel-229-g3fb4ee4
```

Technical Details

OpenAirInterface (OAI) 5G Simulator:
- Runs on Ubuntu 16.04 with low-latency kernel
- L2NFAPI_NOS1 simulator: simulates eNB and UE stacks using nFAPI interface
  - Uses Cisco's Open-nFAPI implementation

Paparazzi UAV Simulator:
- Runs on Ubuntu
- JSBSim Flight Dynamic Model
- Ground Control Station interface
  - Flight plan and aircraft configurations
- Synced with Google Earth

Testing

Tests for the project were performed separately on the OAI and Paparazzi components. Ping tests were used to verify the connection between the eNB and UE OAI components. Additionally, flight patterns were simulated and parameters were adjusted to test Paparazzi.