EE 597 Fall 2019
Distributed Systems Project

Assigned: Nov. 12, 2019
Due: Dec. 5, 2019, on blackboard

1 Introduction to CORE

The Common Open Research Emulator (CORE) is a tool for emulating networks on one or more machines. You can connect these emulated networks to live networks. CORE consists of a GUI for drawing topologies of lightweight virtual machines, and Python modules for scripting network emulation.

Key features of CORE

- Network lab in a box
  - Efficient and scalable
  - Easy-to-use GUI canvas
  - Centralized configuration and control
- Runs applications and protocols without modifying them
- Real-time connection to live networks
  - Hardware-in-the-loop
  - Distributed with multiple COREs
- Highly customizable.

2 Objective

(1) To tinker around with the real-time network emulator tool developed by Boeing and familiarize with the same.

(2) To establish an agreement protocol among the UAVs for each to match to a unique target once the target is within range.

(3) To tweak modules of existing code in Python to increase the performance of communication links between UAVs.

(4) To develop a suitable algorithm in Python for a scenario where drones must complete the mission (i.e., tracking a unique target) efficiently.

(5) To prove the efficiency of the developed algorithm under various scenarios of:

  - Loss rate

For more details, please see PROMPT.txt in uavs_targets.zip.

3 Working Environment

This programming assignment **must compile and run on a standard 64-bit Ubuntu 16.04 system**. To **minimize the chance of hardware incompatibility**, the best way to run Ubuntu Linux is to first install a **virtual machine hypervisor** (Oracle VirtualBox is recommended because it's free and available on both Windows and Mac OS X machines), then install 64-bit Ubuntu 16.04 into a virtual machine inside the hypervisor.

4 Submission Guideline

The code produced by a team will be run through the test script (start_testing.sh) to evaluate the correctness and performance.

The teams are expected to provide their programming assignment which should include

- Any new/modified Python/Shell scripts
- A README.txt file to specify which files are new and how to execute the code

The teams are also expected to produce a document while submitting the project which should include

- A short note on the algorithm used by the group for the agreement protocol
  - A discussion about why this algorithm is better than the existing one
- Some figures/tables to show the performance
  - Some comparisons between the proposed algorithm and the existing one
  - The performance under different scenarios
- Any challenges faced by the team during the entire project (and how you overcame it)

5 Grading Guideline

Programming:

- Code/README.txt (30%)
- Correctness - whether the mission is completed (10%)
- Performance - latency (10%)

Report:

- Algorithm description (20%)
- Result/Comparison (20%)
- Discussion (10%)