# CORNET: A Co-Simulation Middleware for Robot Networks

**Srikrishna Acharya**, Robert Bosch Centre for Cyber Physical Systems, IISc

**Bharadwaj Amrutur**, Robert Bosch Centre for Cyber Physical Systems, IISc

**Yogesh Simmhan**, Dept. of Computational Data Sciences, IISc

**Aditya Gopalan**, Dept. of Electrical Communication Engineering, IISc

**Parimal Parag**, Dept. of Electrical Communication Engineering, IISc

**Himanshu Tyagi**, Dept. of Electrical Communication Engineering, IISc

ROBERT BOSCH CENTRE
for CYBER-PHYSICAL SYSTEMS
Indian Institute of Science, Bangalore-560012, India

# Introduction

Autonomous robots has enormous potential in several domains like search & rescue, surveillance, transport and infrastructure etc.

V2X use-cases: remote-controlled autopilot systems, situation aware driver assistance systems.

Performance evaluation of such networked systems requires appropriate tools

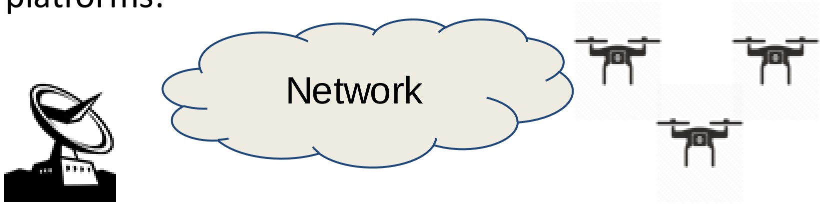
**For example: impact of communication artifacts like packet losses, congestion and routing overheads on the control system of the robots.**

# Problem Statement

Develop a co-simulation framework that integrates simulation of flight dynamics and network related aspects.

The solution should satisfy the following criteria

- Lightweight and scalable to handle complex real-world scenarios
- There should be a common notion of time and position across the two simulation environments and
- Provides APIs to port control algorithms to a real platforms.



# Related work

UAV Simulators :

- human operating flight simulators - Microsoft flight simulator  FlightGear
- autopilot frameworks - PX4 and Ardupilot Software in the loop(SITL)
- Robotic Simulators (3D) - AirSim and Gazebo

Network Simulators:

- commercial network simulators - QualNet and OPNET
- open-source simulators  - NS-3 and OMNET++
- network emulator - Mininet

# Related work

**Joint UAV-Network Simulators :**

**AVENS :** Developed based on X-Plane and OMNET++.

- *HICSS, 2017*

**CUSCUS - ** Developed based on FL-AIR and NS-3.

- *Ad Hoc Networks Volume 68, January 2018, Pages 33-47*

**FlyNetSim -** Developed based on Dronekit SITL and NS-3.

- *MSWIM, 2018*

# Related work

## Application Specific Drone Simulators: Recent Advances and Challenges

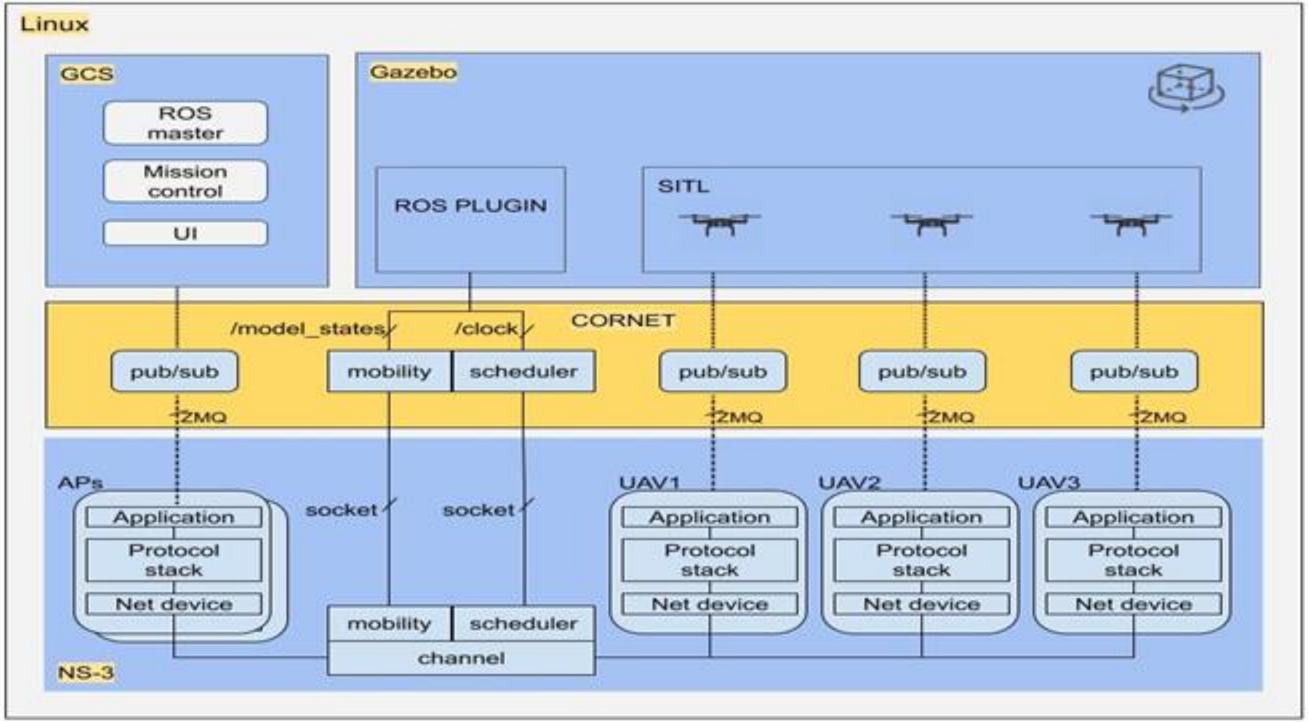
| | Open Source | Network simulation | |
|---|---|---|---|
| HEXAGON | No | No | LabVIEW based GUI, Mathematical Models |
| Simbeeotic | Yes | Limited | Java and uses Maven build system |
| MS Flight Simulator X | No | No | proprietary software by microsoft |
| AirSim | Yes | No | useful tool for AI research |
| AVENS | Yes | Yes | Doesn't provide data path for control, telemetry to and from UAVs |
| UAVSim | Yes | Yes | Focus on UAVNet Attacks and security aspects |
| D-MUNS | No | Yes | NS3 based network simulator |
| ROTORS | Yes | No | AseTec, Gazebo project for multi UAV systems. |

# Our Approach

Our choice of Gazebo with ROS and NS-3 in the Linux environment is primarily driven by

1. Better community support,
2. Gazebo has modular software architecture and extended using plug-ins,
3. NS-3 allows interfaces for external systems, applications and libraries.

# Co-Simulation Architecture

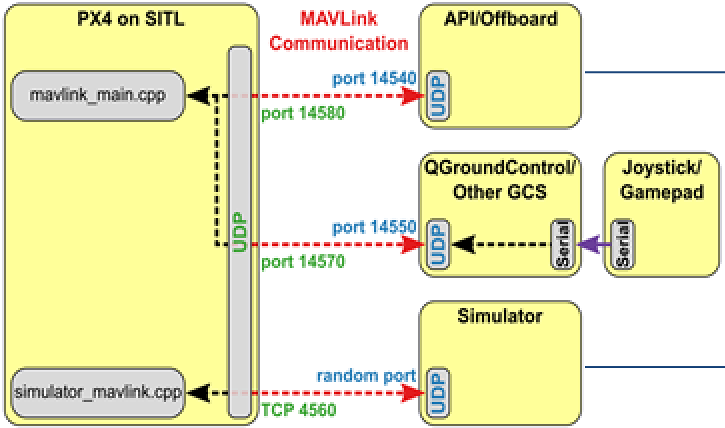


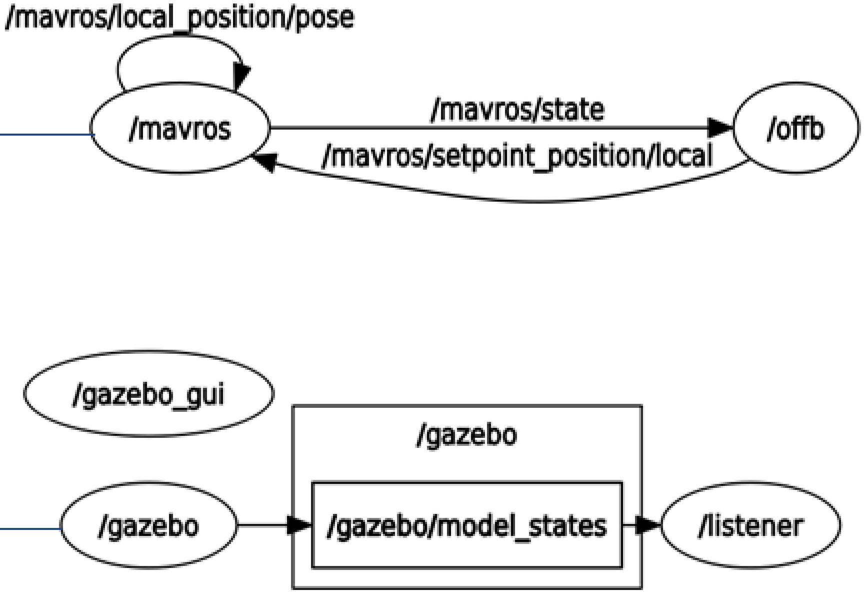
# CORNET Middleware

Middleware interconnect Gazebo and NS-3:

1. ZMQ - end to end data path

1. CORNET ROS plugin - mobility and time sync



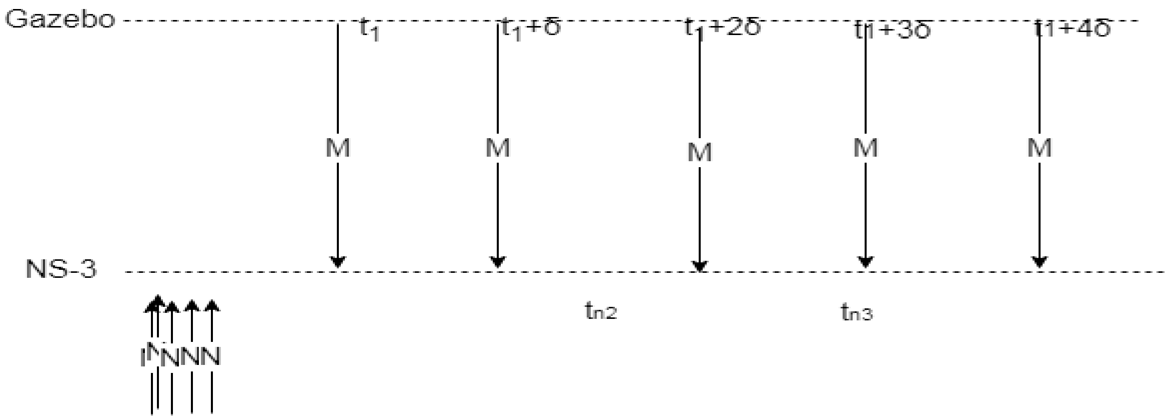



# Why is time synchronization critical

Gazebo simulates physical systems – uses period sampling

NS-3 simulates network systems – deploys event based sampling

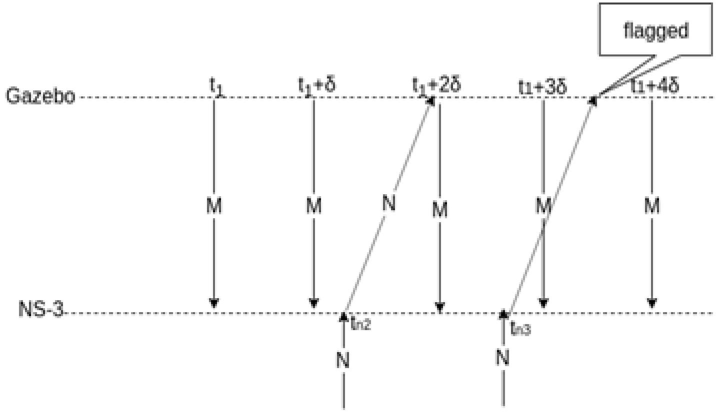
Combination of periodic and event based sampling is not a trivial problem.

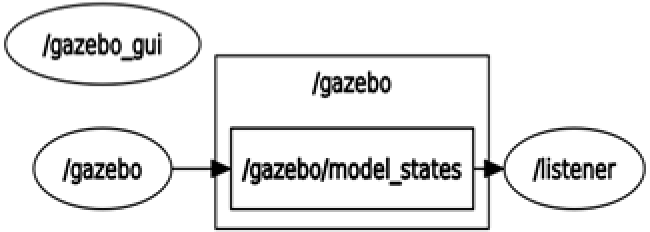


# Time Synchronization

The method we have adopted is referred to as the variable-stepped method:

- Gazebo sim time is reference clock for which NS-3 simulation is tied.
- NS-3 checks for network events and reference clock for execution.
- The middleware checks for the packet leaving NS-3 and releases to Gazebo.
- If the Packet delayed by NS-3 results in delivery after the time lasped in gazebo sim time, we discard these packets.



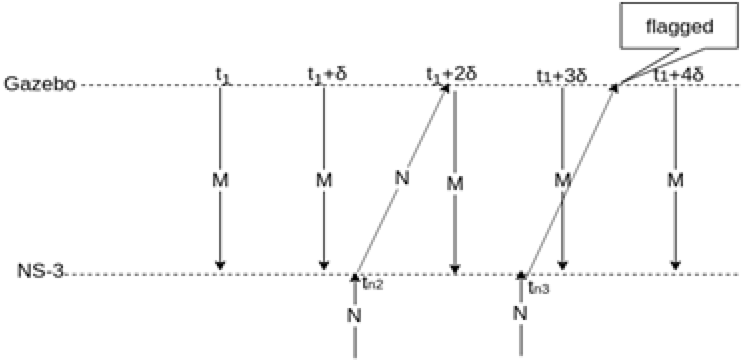



# Criteria for proper functioning

Packet discarding is result of case where NS-3 runs slower than Gazebo simulation.

Ideal scenario is to update the clock of Gazebo to match with NS-3.

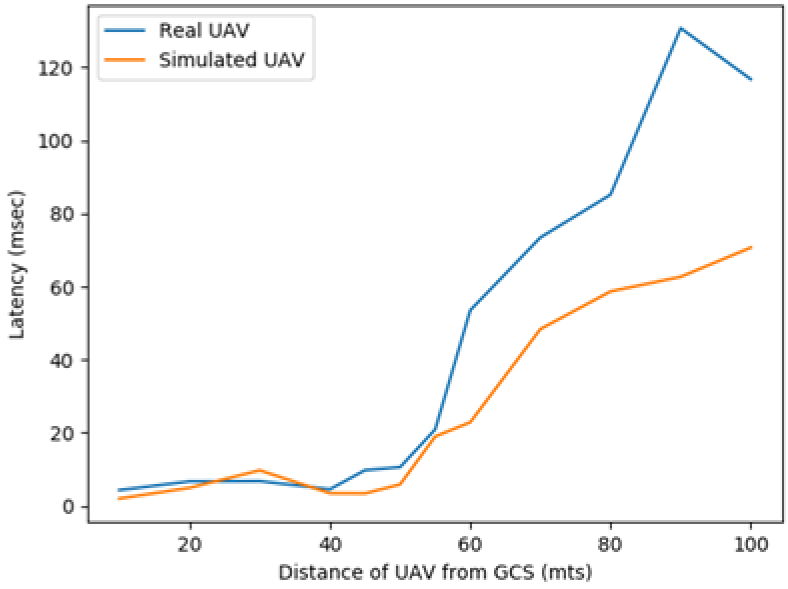
**NS-3 event processing should be faster than Gazebo**, *which is the case in normal execution scenarios.*



# Evaluation

Offboard control of Single UAV:

We measure end-to-end network latency for both the real world and the simulation

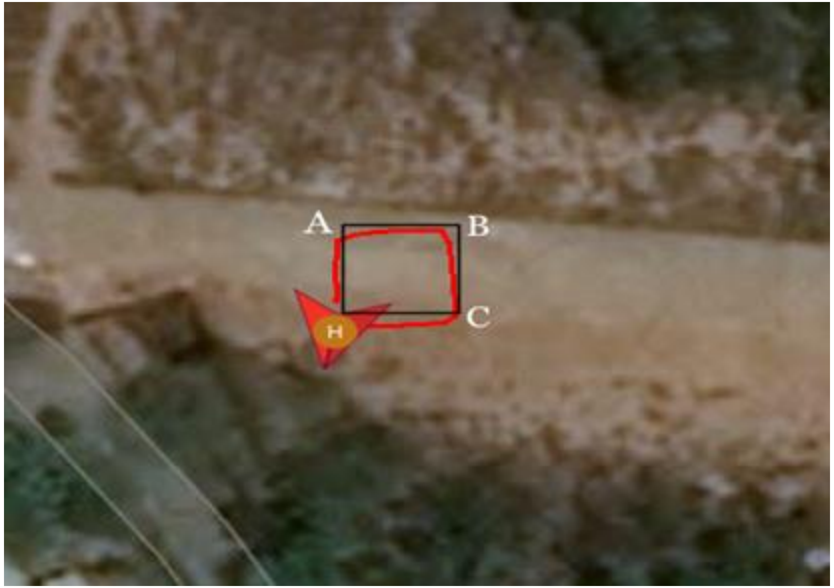


# Evaluation

Fixed Trajectory Control:

GCS sends commands to the UAV to follow a pre-determined trajectory.

This is a way-point based navigation with the controller present in the GCS




Simple evaluation of the framework's ability to develop and test new network functions for UAVNETs.

# Comparison Table

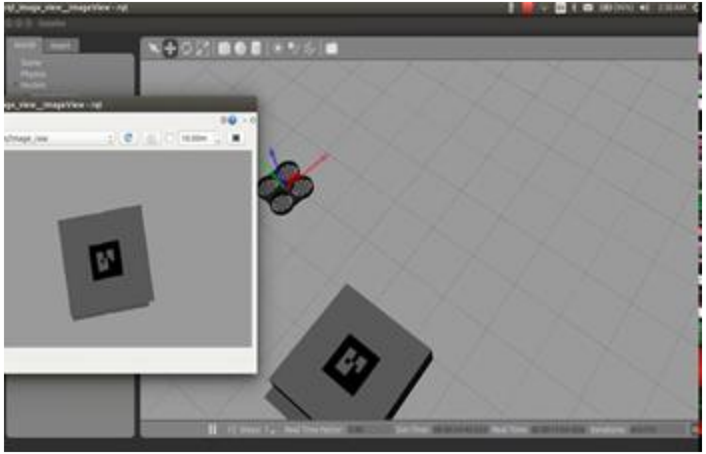
| | FlyNetSim | CUSCUS | AVENS | CORNET |
|---|---|---|---|---|
| Network simulator | NS-3 | NS-3 | OMNET++ | NS-3 |
| Physics simulator | Dronekit SITL | FL-AIR | X-Plane | Gazebo |
| Mobility | Deep packet inspection | Shared memory | XML based | ROS topic |
| Data-path | ZMQ | Ns-3 bridge | - | ZMQ |
| Time Sync | Yes | No | No | Yes |

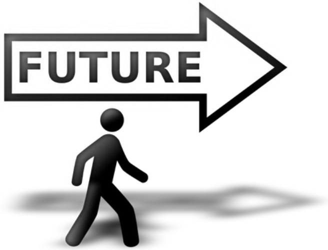
# Future Work

Modeling latency requirements for Droneport to control drone to enable precise landing,

Extend the framework to support LTE and 5G to enable multi-technology networking,

Evaluate routing algorithms for UAVNETs specifically position-based routing protocols.





# Acknowledgment

We would like to thank,

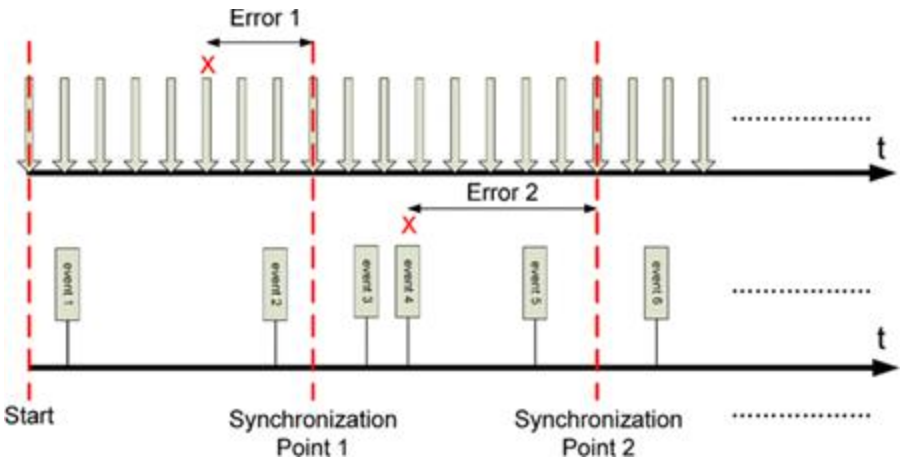
IISc’s 5G-V2X group, Department of Telecom, Govt of India

Indian National Academy of Engineers for their constant support.

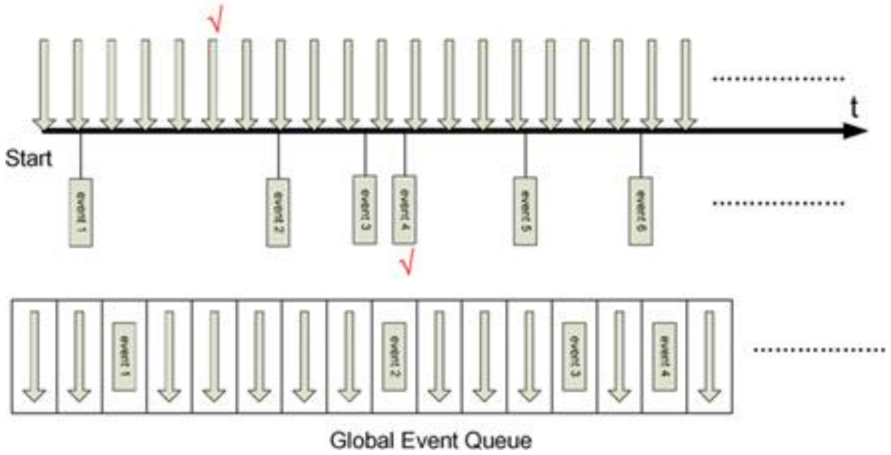
We are grateful to our project staff, especially Varun, who provided expertise in Gazebo simulation that greatly assisted the research.

**Questions ?**

# Time Synchronization




Time-stepped method




Global-event driven method