FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Experimental Evaluation Platform for Resilience in CPS

Will Emfinger, Gabor Karsai, Pranav Kumar
Vanderbilt University

Berkeley UNIVERSITY OF CALIFORNIA
Massachusetts Institute of Technology
NSF
MICHIGAN
VANDERBILT UNIVERSITY

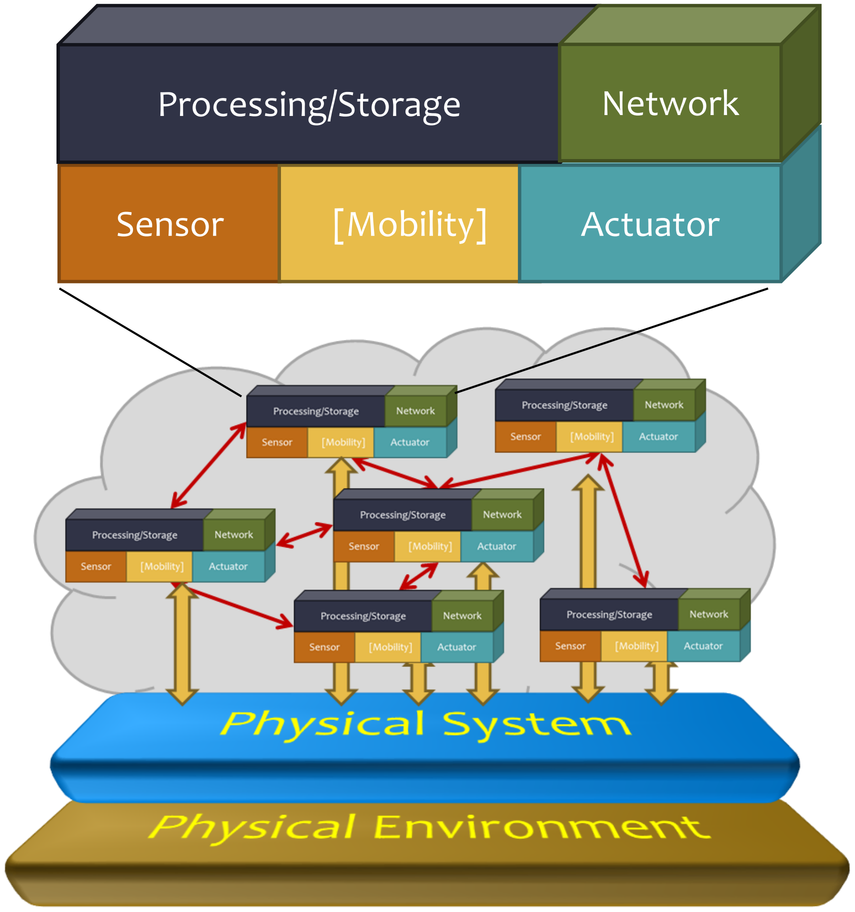
Overview

* Goals
* Design
* Experiments
* Conclusions

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Goals – Evaluating CPS Resilience

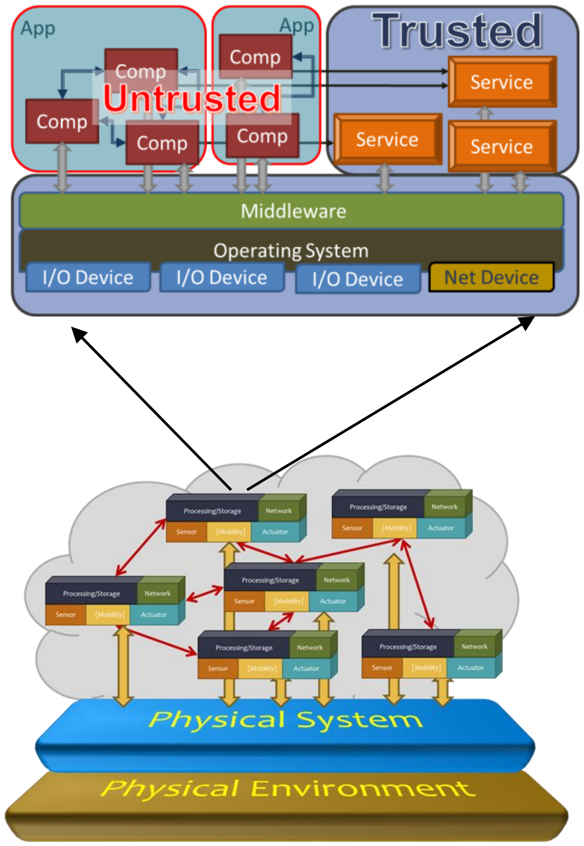
- Resilience is a system-level property, permeating the entire CPS architecture
  - → Must be evaluated in the *system* context
- CPS Software is *not* constant, it is continuously updated and extended
  - → Must rely on a *software platform* that provides core abstractions and services for the applications
- *Interactions* across and within the physical and cyber domains is a key feature of CPS
  - → Must be recognized and managed throughout the system's lifecycle



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Goals – Evaluating CPS Resilience

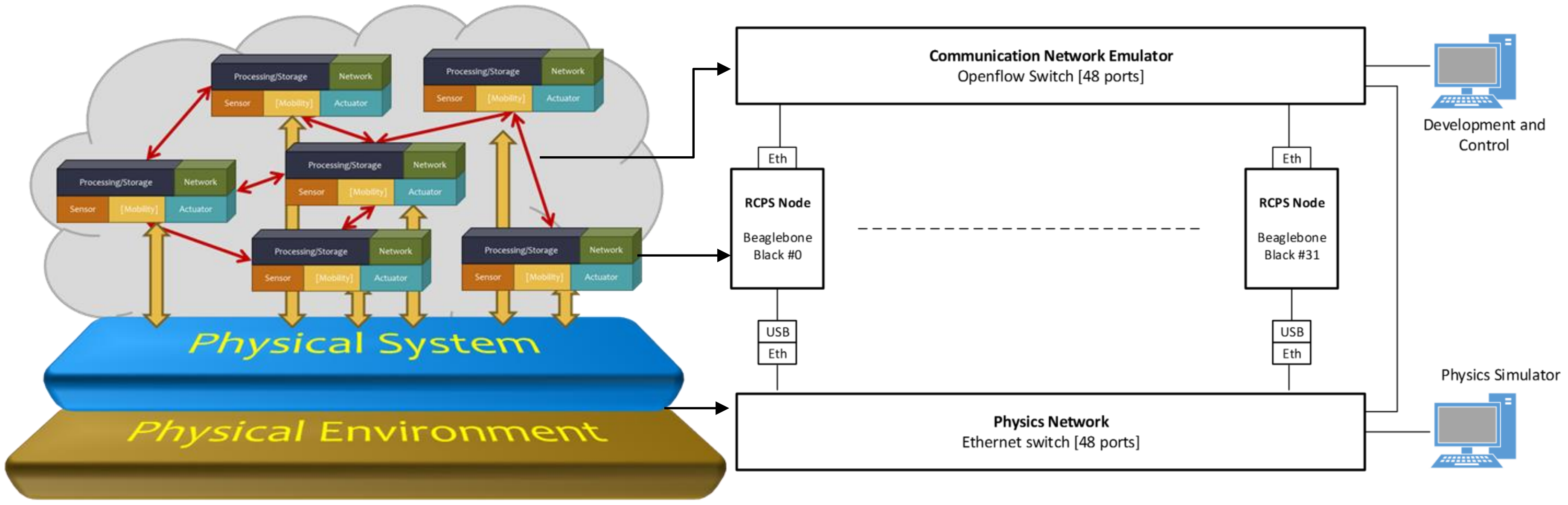
* CPS Software Model:
  * Trusted (dependable, protected) *platform* + Untrusted *apps (components)*
* Challenges:
  * How to model the resilient architecture? What makes it resilient?
  * How to build a resilient software application platform for CPS?
  * How to analyze in a scalable manner to obtain assurances for resilience?



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

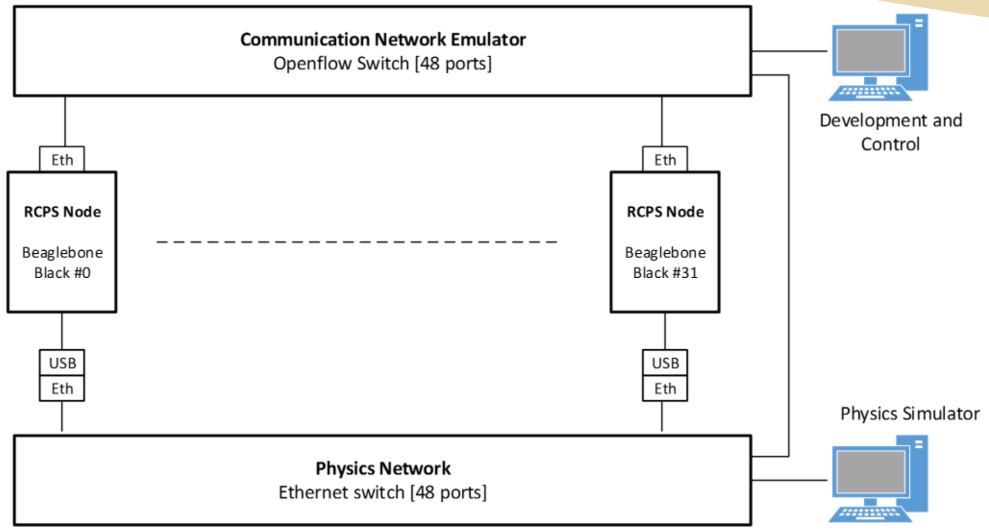
Design: Testbed

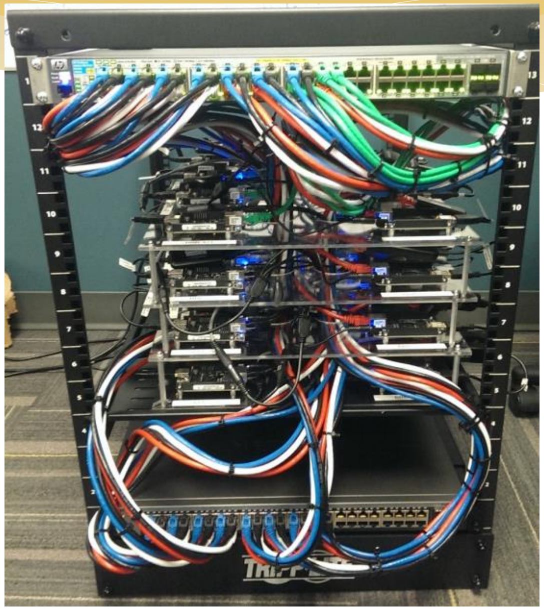
- RCPS Node: embedded computer, running the core platform software, executes applications, communicates with other RCPS nodes, interacts with the (simulated) physical environment
- Communication Network Emulator: facilitates network flows between the nodes, flows can be controlled and restricted (bandwidth limitations, communication dropouts, etc.)
- Physics network: provides connectivity between the RCPS nodes and the physics simulator



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Design: Testbed



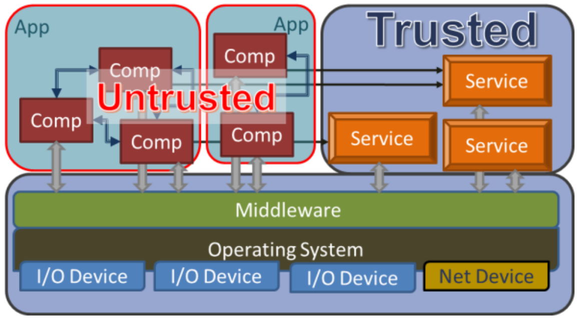


* Physics Simulator: simulates the physical platform, the attached sensors and actuators, and the external environment. RCPS nodes receive data from the simulated sensors and send commands to the simulated actuators. May also send relevant data to network emulator to change network performance on the fly.
* Development and Control: a desktop machine for all development and experiment control activities: building applications, uploading applications to the RCPS network nodes, running and controlling experiments.

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Design: Software Infrastructure

- Resilient CPS (Software) Platform
  - Component-based application model: *component model* with interaction semantics
  - Application *deployment model*: trusted and managed deployment
  - Resource *monitoring* and constrained *information flows* on the platform level
  - Hardened *platform interfaces* and *services* (for resilience)

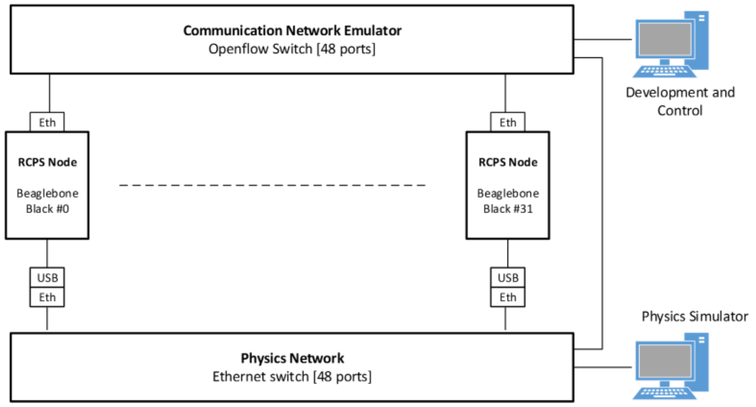


ROSMOD: An Extension to the Robot Operating System (ROS) – work in progress
https://github.com/rosmod

P. Kumar,W. Emfinger, A. Kulkarni, G. Karsai, D.Watkins, B. Gasser, C. Ridgewell, and A. Anilkumar. ROSMOD: A Toolsuite for Modeling, Generating, Deploying, and Managing Distributed Real-time Component-based Software using ROS. In Proceedings of the IEEE Rapid System Prototyping, RSP 2015, Amsterdam, Netherlands, 2015. IEEE

Experimentation

Resilience to…	Realize it in ….
Physical plant faults	Physics simulator
Sensor/actuator faults	Physics simulator
Unforeseen environment	Physics simulator
Cyber effects on controller	RCPS node
Cyber effects on network	Network emulator



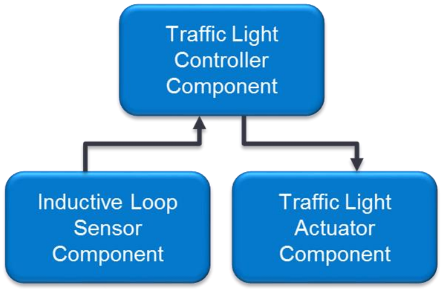
Assumptions:

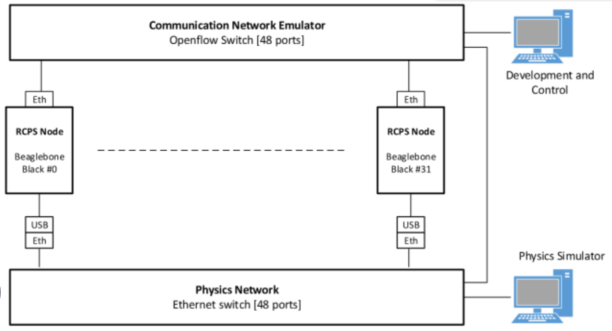
- Physics simulator is timing accurate w.r.t dynamics, can realistically simulate faults, has sufficient I/O bandwidth
- Network emulator can realistically emulate network conditions (limited bandwidth, latency, jitter, packet drops) and cyber effects on the network (DDOS, protocol attacks, etc.)
- RCPS node running embedded control software is realistic; including I/O behavior

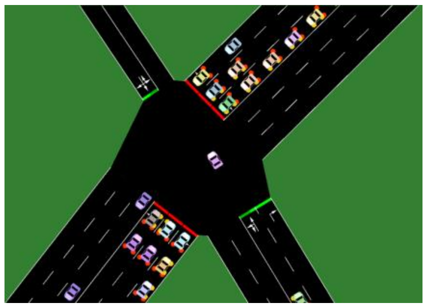
FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Example: Traffic monitoring and control

- Problem: How to evaluate resilient monitoring and control schemes for traffic lights where the communication network or the sensor nodes are under cyber attacks?
- Simulation: SUMO (open source traffic simulator)
- RCPS node: Monitoring and controller code
- Cyber attacks
  - Sensors: simulate it in SUMO (spoofing)
  - Network (DDOS): emulate it in the network emulator – or – use a few RCPS nodes as traffic generators to emulate packet flooding


Traffic Light Controller Component
Inductive Loop Sensor Component
Traffic Light Actuator Component

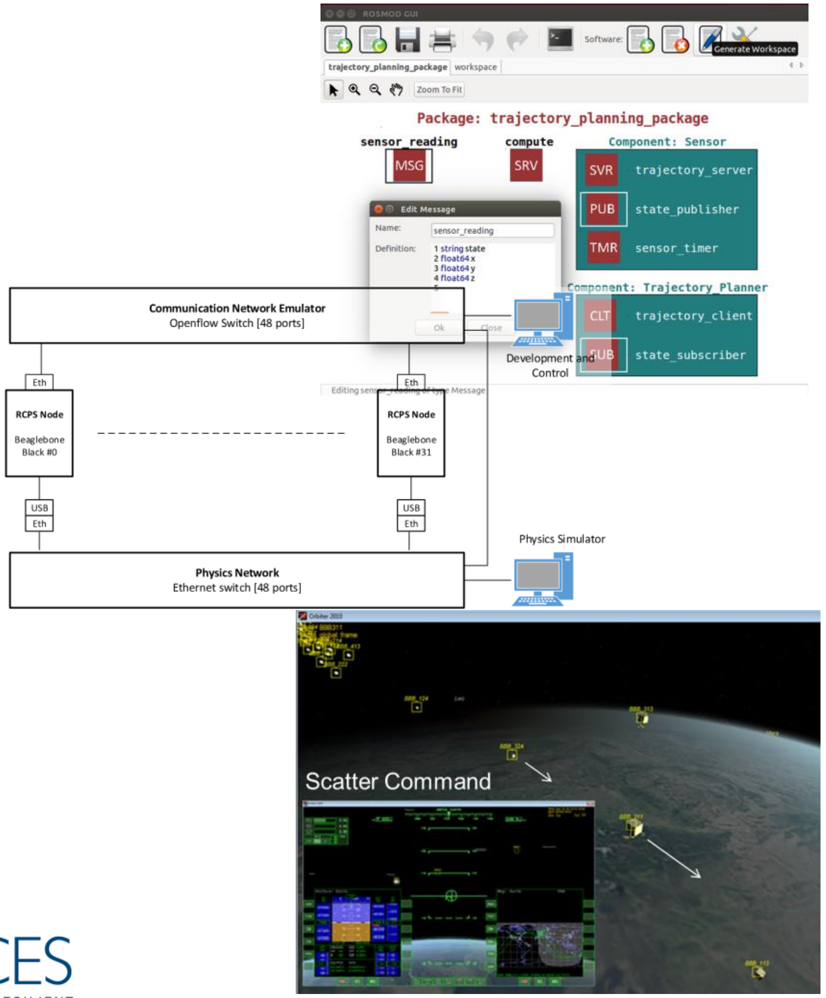




FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Example: Spacecraft flight control

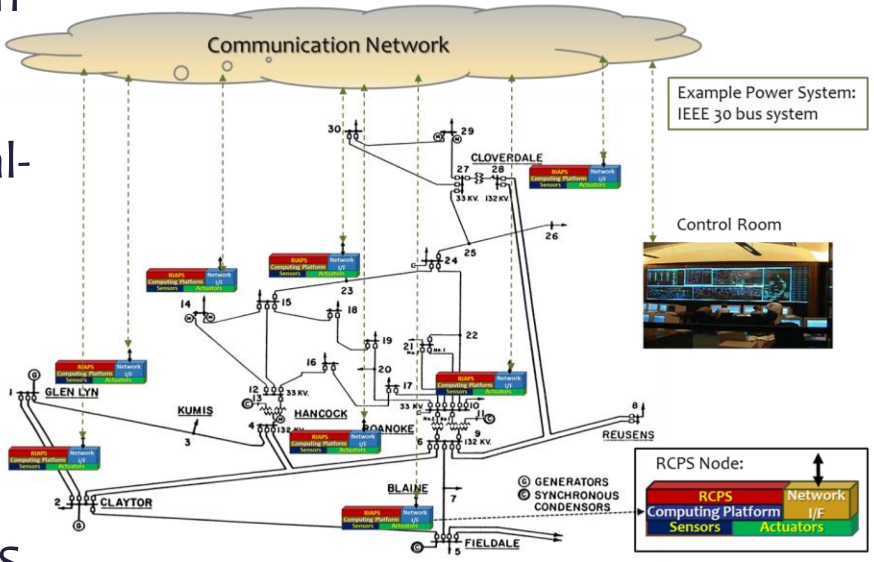
- Problem: How to test the robustness of communication protocols and control algorithms (e.g. trajectory planners) for a cluster of satellites flying in formation, where the network may be degraded?
- Simulation: Orbiter (open source spaceflight simulator)
  - Calculates distance between satellites, informs network emulator to accurately model bandwidth effects
- RCPS node: Monitoring and controller code
- Cyber attacks
  - Network (jamming): emulate it in the network emulator



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Example: Power transmission/distribution

- Problem: How to evaluate a distributed state estimation / remedial action scheme for a power transmission system that runs on a distributed software platform?
- Simulation: PowerSim (MATLAB) or Opal-RT
  - Timing accuracy is necessary
- RCPS node: Monitoring and distributed controller code
- Cyber attacks
  - Network (DDOS): emulate it in the network emulator – or – use a few RCPS nodes as traffic generators to emulate packet flooding



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Summary / Future work

- RCPS Testbed provides a platform for experimentation with resilience in a realistic setting
  - Computing hardware for embedded controller : real
  - Network : emulated through a OpenFlow switch
  - Physics: simulated on a dedicated physics engine
  - Cost = ~$5K ; Design is public and software is open source
- Experimentation with physical faults and cyber effects
  - Data logging ; results are post processed
- Next steps:
  - Document all aspects of the testbed
  - Improve the management interface
  - Conduct experiments
  - Investigate how simulation performance can be improved
    - Parallelized simulation / C2WT/HLA-based distributed simulation

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS