A Testbed to Simulate and Analyze Resilient Cyber-Physical Systems

Pranav Kumar, William Emfinger and Gabor Karsai
Institute for Software Integrated Systems,
Dept. of EECS, Vanderbilt University,
Nashville, TN 37235, USA
Email:{pkumar, emfinger, gabor}@isis.vanderbilt.edu

***Abstract*—We have created a testbed for the development and testing of Cyber-Physical Systems (CPS) applications. This testbed incorporates smart network hardware which allows for high-fidelity emulation of the system's network characteristics, and simulation systems which allow for high-fidelity simulation of the CPS, its environment, its sensors, and its actuators. We discuss the architecture of this testbed, the types of experiments and applications which can be run on the testbed, some of the testbed's limitations, and some extensions to the testbed.**

I. Introduction

CPS are hard to develop hardware/software for; because the software is coupled with the hardware, software testing and deployment may be difficult. Many systems require rigorous testing before final deployment, but may not be able to be tested easily in the lab or in the real world without first providing the assurances that the tests produce. For such systems, a closed-loop simulation testbed is necessary which can fully emulate the deployed system, including the physical characteristics of the nodes, the network characteristics of the systems, and the sensors and actuators which the systems used. Furthermore, many of these systems use specialized embedded computers which have very different software and hardware support than cloud-based testing infrastructure can provide.

use automotive industry testbed as example, this is cost effective version for other embedded systems developers and researchers

architectural description: require good physics integration -¿ distributed -¿ time sync -¿ low jitter -¿ accurate timing what kind of tests can you run? what kind of systems is this good for?

limitations: can't do all integration testing usb-ethernet -¿ limitation for physics possibly -¿ what scenarios switches can be limitations depending on applications and such -¿ to what can they support? processing limitations -¿ dependent on testbed -¿ hardware prototyping boards should be used for relevant hardware how does the system/testbed scale: switches? openflow?

future work: test/measure jitter

II. Problem Statement

III. Testbed

A. Overall Architecture

1) Application Network and Emulation:

2) Physics Network:

3) Physics sim and interface:

4) Actual hardware:

B. Testing

1) Orbiter: Good for certain types of system testing but not great for doing tests across multiple domains

2) KSP: Though not as accurate, allows testing between multiple interacting domains.

IV. Future Work

V. Related Research

VI. Conclusions

Acknowledgment

This work was supported by ...