Developing and Validating a Generic Environment for Real-Time Networking Experiments

Filip Rezabek¹, Marcin Bosk¹, Thomas Paul¹, Kilian Holzinger¹, Sebastian Gallenmüller¹, Angela Gonzalez², Abdoul Kane², Francesc Fons², Zhang Haigang², Georg Carle¹, and Jörg Ott¹

**Motivation**

Provide a structured approach that enables computer network assessment with experiments which are:

- Accurate and realistic
- Scalable and highly configurable
- Reproducible and repeatable

Various approaches already exist, e.g., utilizing:

- Simulation – highly repeatable and configurable; may be unrealistic omitting real deployment artifacts
- Hardware – more realistic; varying levels of repeatability and possibly uses proprietary solutions

**Can we combine benefits of simulation and hardware approaches?**

Introducing EnGINE [1,2] experimental environment:

- Accurate, reproducible, and scalable experiments
- Based on commercial off the shelf (COTS) hardware and open-source software
- Configuration flexibility via Ansible playbooks
- A stepping stone towards a generalized solution

**Hardware Deployment**

15 highly interconnected nodes acting as ZGWs/VCCs

COTS hardware and NICs with TSN support:

- Intel i210 – 1Gbit/s: IEEE 802.1 AS, Qov, Obv
- Intel i225 – 2.5Gbit/s: IEEE 802.1 AS, Qov, Obv
- Intel i350 – 1Gbit/s: IEEE 802.1AS
- Intel x552 – 1Gbit/s: IEEE 802.1AS

Sensors:

- LiDAR – Livoxtech Mid 40
- 2 Cameras – Reolink Full HD

**Design Overview**

Experiments defined based on three components:

- Input – defines the experiment
- Network Processing – encompasses the tested system and hardware
- Output – recorded experiment results and artefacts

**Experiment Orchestration**

Orchestration accomplished using Ansible.

Experiments conducted in four phase campaigns being defined using five configuration files.

In Install and Setup phases prepare the environment.

Scenario phase conducts the individual campaign experiments in a loop.

Process prepares combined results of a campaign.

**Node Management**

Nodes managed using the POS framework [3] combined with Ansible and Bash scripts.

Individual tasks executed according to five campaign configuration files defining which Ansible playbooks need to be executed on the nodes.

During the actual execution of the experiment, tasks controlled locally on each node.

After all experiments are finished, the management host collects the results and evaluates them.

**Framework Validation**

Initial focus on IVN and TSN requirements:

- Strict delay bound of 2ms across 7 hops
- Strict jitter bound of 100µs across 7 hops

Validation performed without and with TSN standard (IEEE 802.1 Qav) enforced on the nodes

**Future Work**

Generalize framework’s capabilities beyond Layer 2 Ethernet and TSN:

- Include Layer 3 configurations
- Consider novel layer 3 protocols
- Consider various Transport Layer protocols
- Include application-level solutions
- Include more complex traffic generation capabilities
- Enhance experiment types to embrace specialized hardware and software
- Include proprietary solutions
- Include other specialized NICs

Consider simulation capabilities to enhance the hardware solution – Digital Twin

Improve result evaluation capabilities

---

¹Chair of Connected Mobility
²TUM School of Computation, Information and Technology
²Technische Universität München

**Acknowledgments**

This work has been supported by the German Research Foundation (DFG) under grant number 423760251, and partially funded by the European Union (European Regional Development Fund - ERDF) through the German Federal Ministry of Education and Research (BMBF) under contracts 01IS20047A and 01IS20049A.