HAW HAMBURG

Requirements Analysis and Performance Evaluation of SDN Controllers for Automotive Use Cases

2020 IEEE Vehicular Networking Conference (VNC)
December 16–18, 2020 | Virtual Conference

Randolf Rotermund, Timo Häckel, Philipp Meyer, Franz Korf, and Thomas C. Schmidt
Dept. Computer Science, Hamburg University of Applied Sciences, Germany

Contact: randolf.rotermund@haw-hamburg.de

secVI

Outline

I. In-Vehicle Networks
II. Software Defined Networking
III. Requirements Analysis
IV. Performance Evaluation
V. Controller Applications
VI. Conclusion and Outlook

I.

In-Vehicle Networks

Present State of In-Vehicle Networks

- Functions and services through Electronic Control Units (ECUs)
- Different proprietary bus systems (CAN, MOST,...)
- Domain based network

Emerging In-Vehicle Networks

- Increased dynamic network traffic
  - Internet connectivity
  - Communication with environment (Vehicle-to-X)
- Switched Ethernet-based backbone
- Time-Sensitive Networking (TSN)

- Software Defined Networking (SDN) approach for dynamic network traffic

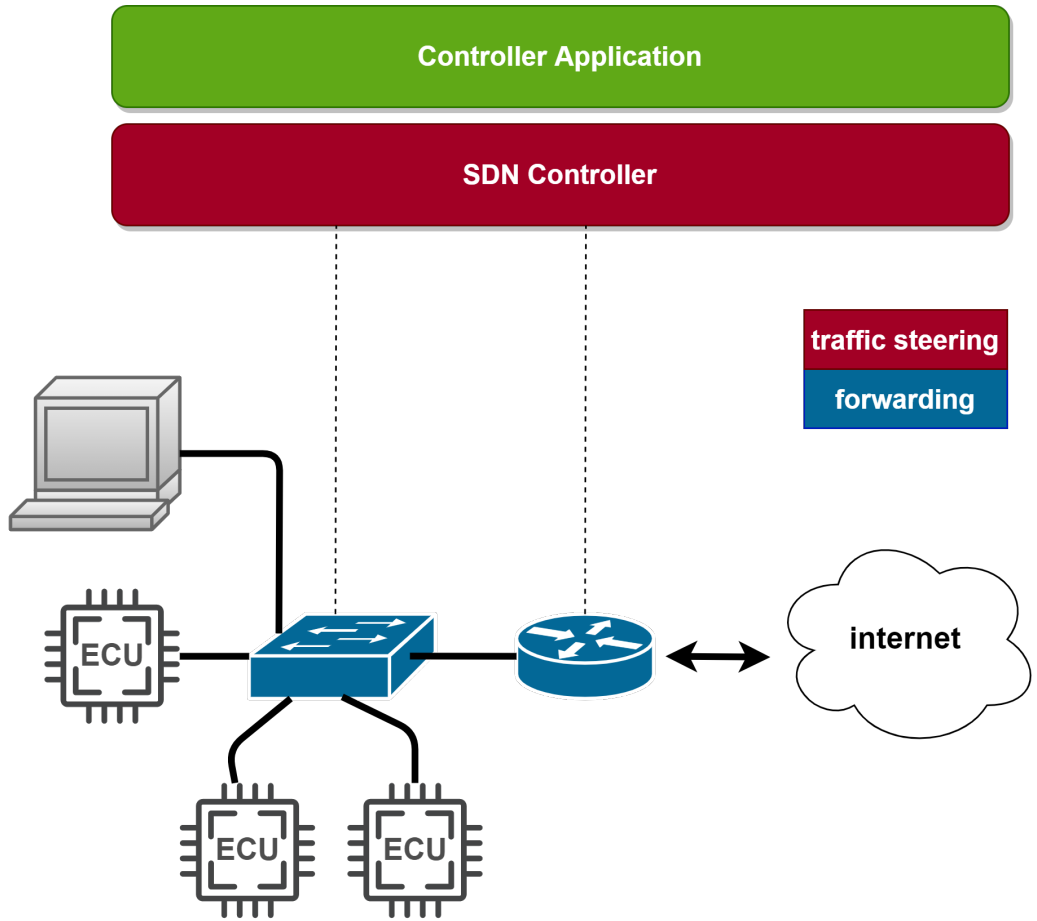
II.

Software Defined Networking

Software Defined Networking

- Logically centralized
  - SDN controller (Software)
  - Forwarding devices (Hardware)

- SDN controller
  - Global view on network
  - Steer network traffic
  - Development of controller applications



Research Questions

1. What requirements ?
2. How do SDN controllers perform ?
3. Can requirements be fulfilled by applications ?

1. Requirements analysis
2. Performance evaluation
3. Controller application examples

III.

Requirements Analysis

Requirements

Real-time

- Quality of Service
- Scheduled configuration
- Short Start-up time

Safety

- Link failure detection
- Transaction
- Controller Redundancy

Security

- Access control
- Network statistics

Remaining functional requirements

Remaining non-functional requirements

Evaluation Categories

- **Feature:** Function provided by the controller
- **Performance:** Must be measured in the performance evaluation
- **Application:** Can be fulfilled by a controller application

Candidate Selection Based on Feature Requirements

- Over thirty SDN controllers
- Selection of controllers by provided feature
- No single controller provides all wanted feature
- Only one feature not fulfilled by any controller
- Most promising controllers tested in performance evaluation

Controller Candidates

ONOS

- Java
- Open-Source

OpenDaylight (ODL)

- Java
- Open-Source

Lumina

- Java
- Proprietary
- Based on ODL

Ryu

- Python
- Open-Source

OpenMUL

- C/C++
- Open-Source

IV.

Performance Evaluation

Metrics

- Metrics are related to requirements

Quality of Service
- Asynchronous Message Processing Time

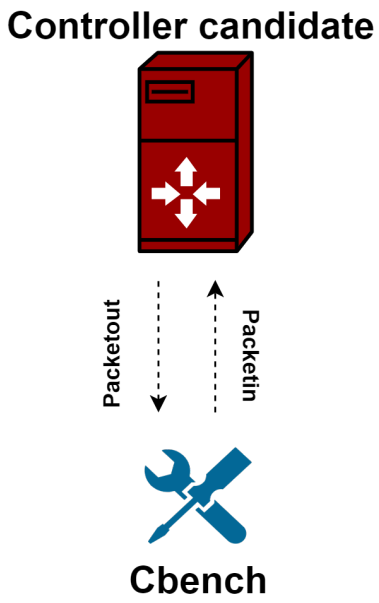
Short start-up time
- Start-Up time

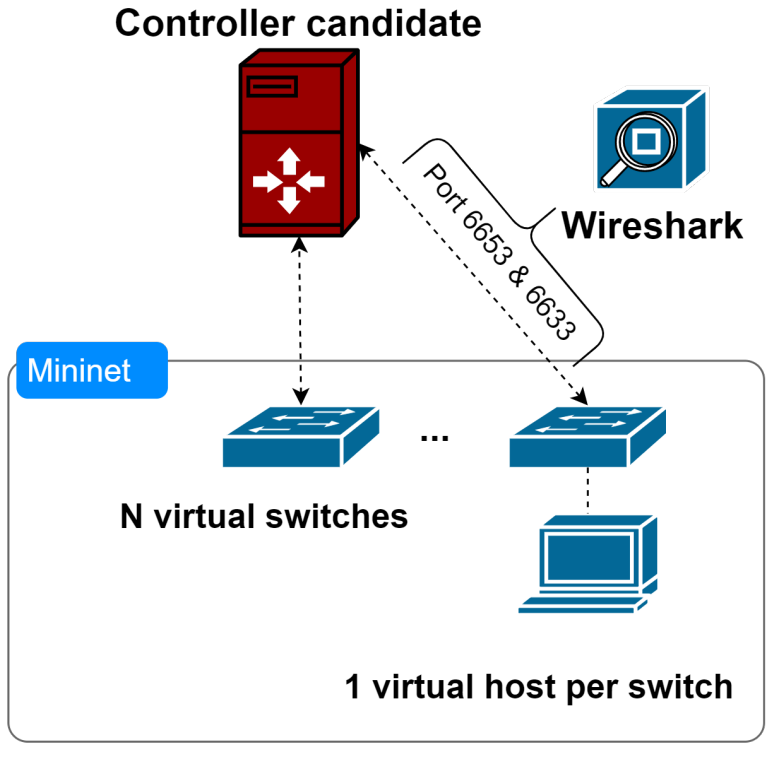
Link failure detection
- Topology Change Detection Time

Controller redundancy
- Failover Time

Testbed

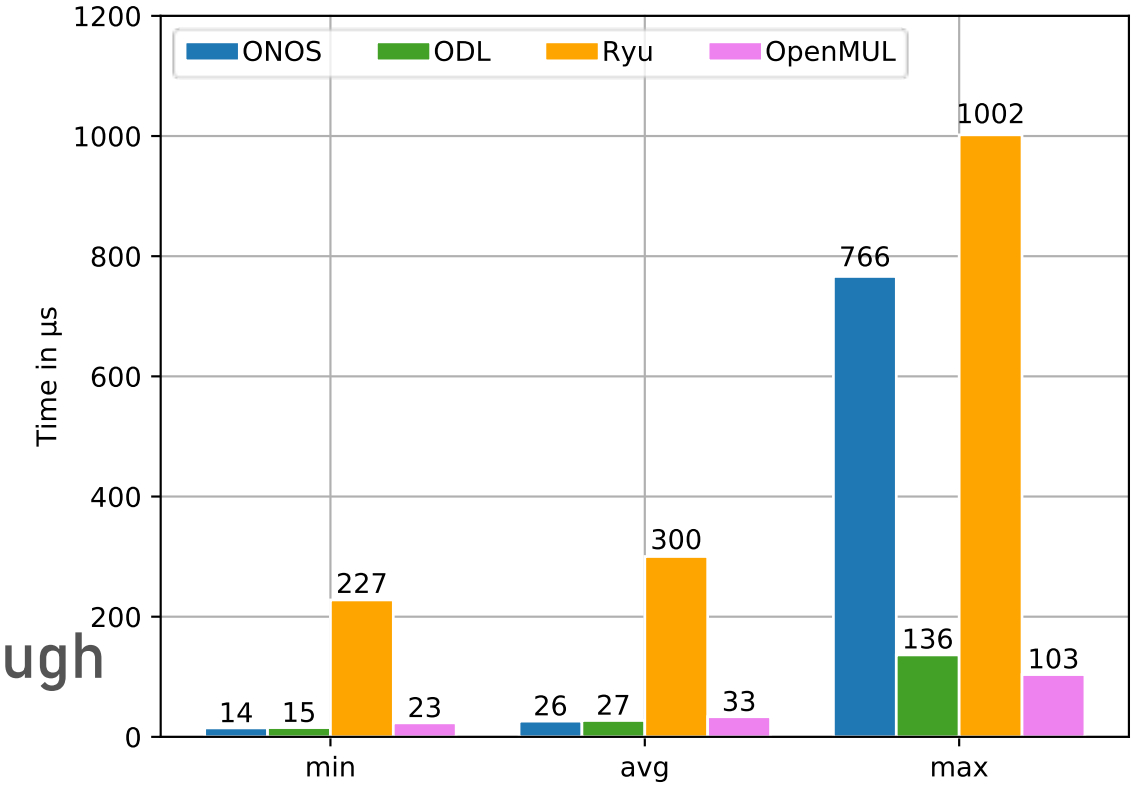
- Cbench: benchmarking application
- Mininet: network emulation
- Wireshark: packet capture





Quality of Service

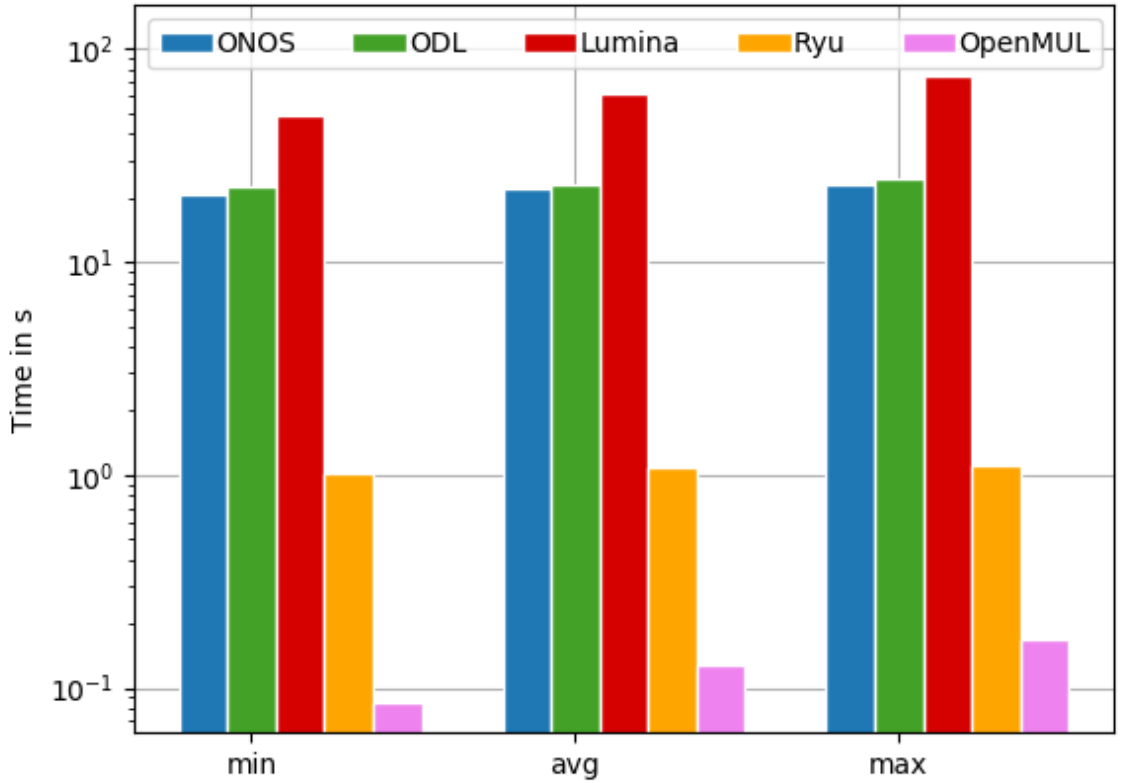
- Asynchronous message processing time
- Delay
  - arrival of asynchronous message
  - outgoing controller message
- Maximum end-to-end delay
  - 10ms control data
  - 150ms multimedia data
- Maximum processing time is low enough
- Extension for guarantees is needed



Short Start-Up Time

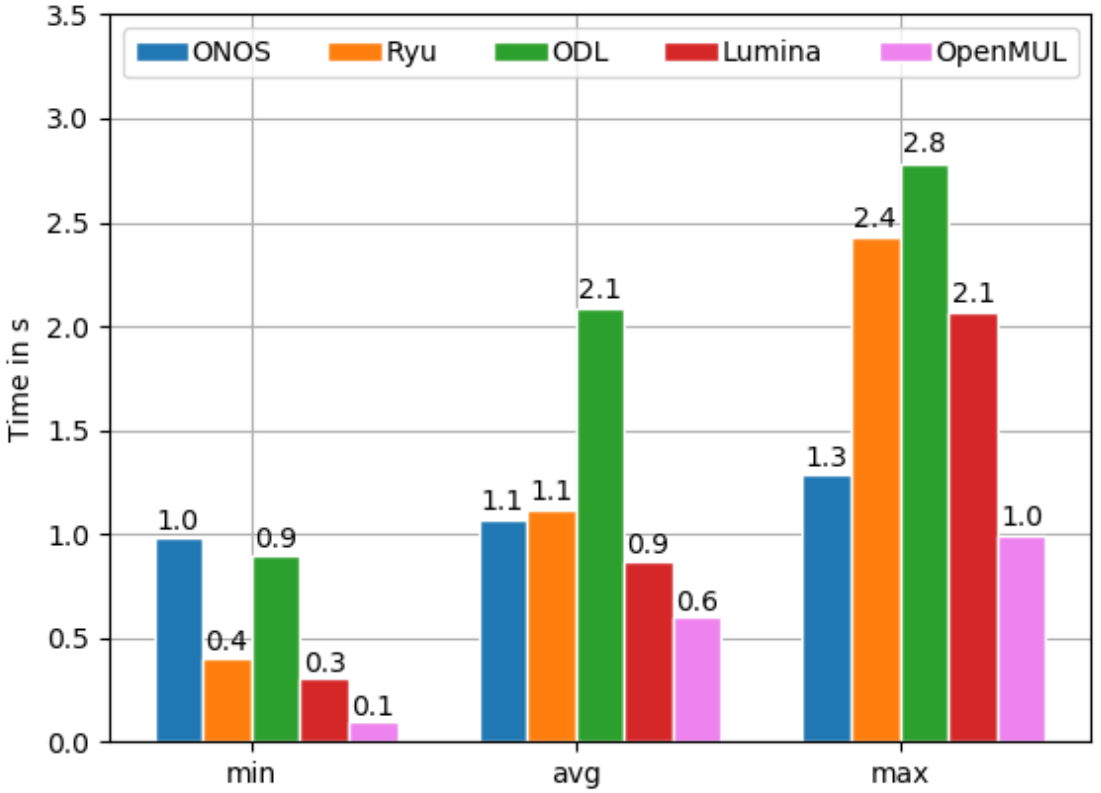
- Start-Up time
- Delay
  - Start of controller
  - Operational state

- Todays ECUs start-up time is under 200ms
- Except OpenMUL all needed multiple seconds
- C-based controller



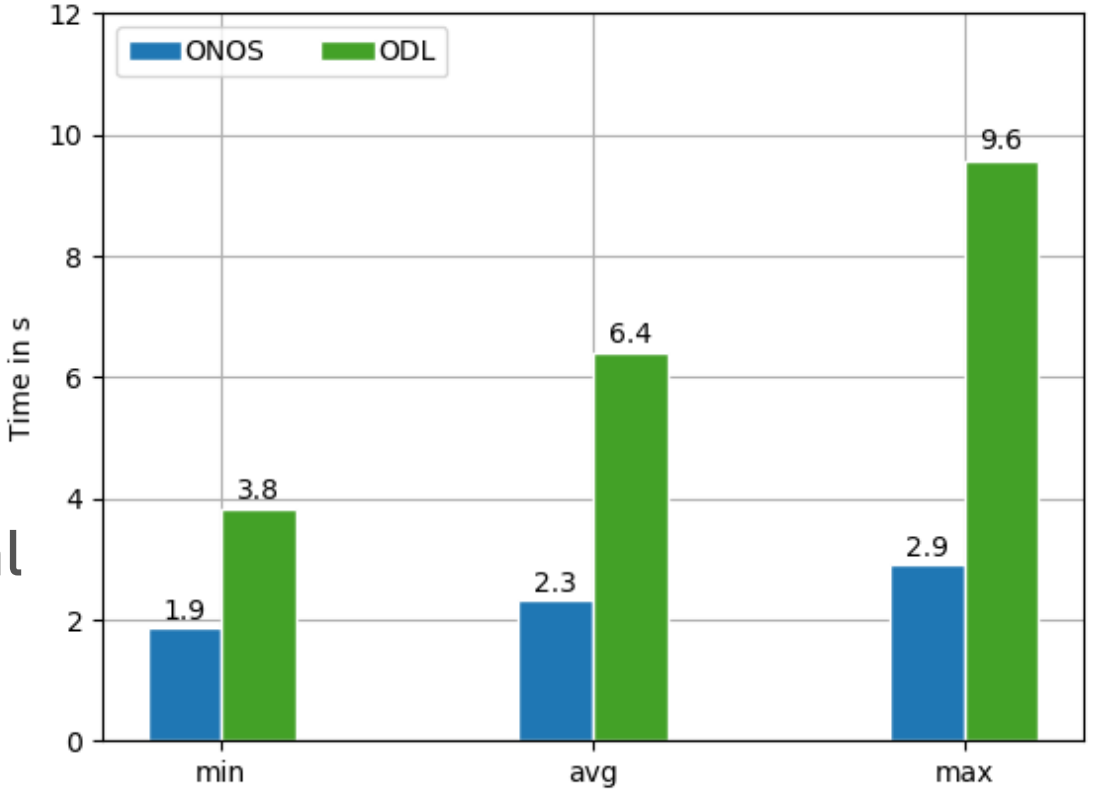
Link Failure Detection

- Topology change detection time
- Delay
  - Topology change
  - Detection
- Relevant for networks with redundant connections
- Maximum end-to-end delay
- Too high to be fallback strategy



Controller Redundancy

- Failover time
- Delay
  - Failure of controller instance
  - Replacement
- Important for management of safety critical traffic
- Failover time too long for safety critical traffic
- Pre-configured forwarding devices



V.

Controller Applications

Implementation Platform

- Requirements with controller applications
- Automotive specific applications
- ONOS
- Realistic in-vehicle network prototype

2016‘ Seat Ateca Prototype

Controller Application Examples

Quality of Service

- Priority Queues

Scheduled configuration

- Configuration with timestamps

Transaction

- Configuration list

Access control

- Access control list

Network statistics

- Report collection

VI.

Conclusion and Outlook

Conclusion And Outlook

- SDN controllers must undergo a redesign
  - Automotive specific features
  - Embedded system compatible
- Extensions are needed to boost performance
- Fulfilling requirements with controller applications

Future Work

- Development of automotive specific controller implementation
- Development of controller applications

Acknowledgements

This work is funded by the German Federal Ministry of Education and Research (BMBF) within the SecVI project.

SPONSORED BY THE

Federal Ministry of Education and Research

secvi.inet.haw-hamburg.de