Towards a Shared Evaluation Environment for Software-Defined-Networking Applications

Fachgespräch Network Softwarization
Addis Dittebrandt, Michael König, **Felix Neumeister** | October, 13th 2017
Central Question

How are SDN-applications evaluated and how can this evaluation process be simplified?
Outline

- Challenges of simulative SDN-Application Evaluation
  - Poor Reproducibility of Results
  - Comparing of Results often not possible
  - Usage of Simulators unnecessarily complicated
- Approach and Implementation
- Usage
- Conclusion
Poor Reproducibility of Results

Problems hindering reproducibility:

- Conflicting experiment descriptions
- Unclear parameters
- Broken artifacts

⇒ What were the exact experiments done?

Approach: Allow and require explicit experiment description
Poor Reproducibility of Results

Problems hindering reproducibility:
- Conflicting experiment descriptions
- Unclear parameters
- Broken artifacts

⇒ What were the exact experiments done?

Approach: Allow and require explicit experiment description
Poor Reproducibility of Results

Problems hindering reproducibility:

- Conflicting experiment descriptions
- Unclear parameters
- Broken artifacts

⇒ What were the exact experiments done?

**Approach:** Allow and require explicit experiment description
Comparing of Results often not possible

What makes results incomparable:
- Unclear description of simulated scenarios
- Broadly similar scenarios with different parameters
- Specific description often not given
- No common ground on realistic scenarios

⇒ How can a simulation scenario be fully described?

Approach: Specify a format to describe a simulated scenario in a single file
Comparing of Results often not possible

What makes results incomparable:

- Unclear description of simulated scenarios
- Broadly similar scenarios with different parameters
- Specific description often not given
- No common ground on realistic scenarios

How can a simulation scenario be fully described?

Approach: Specify a format to describe a simulated scenario in a single file
Comparing of Results often not possible

What makes results incomparable:

- Unclear description of simulated scenarios
- Broadly similar scenarios with different parameters
- Specific description often not given
- No common ground on realistic scenarios

⇒ How can a simulation scenario be fully described?

Approach: Specify a format to describe a simulated scenario in a single file
Configuration of Simulators complicated

Configuration time-consuming and error-prone:
- Can induce side effects into results
- Configuration efforts duplicated

Current workflow when using simulators:
- Familiarize with simulator API
- Model topology and traffic
- Implement topology and traffic in simulator
- Configure simulation environment with external components

How can problems be mitigated?

Approach: Facilitate easy setup using shareable configuration files
Configuration of Simulators complicated

Configuration time-consuming and error-prone:
- Can induce side effects into results
- Configuration efforts duplicated

Current workflow when using simulators:
- Familiarize with simulator API
- Model topology and traffic
- Implement topology and traffic in simulator
- Configure simulation environment with external components

⇒ How can problems be mitigated?

Approach: Facilitate easy setup using shareable configuration files
Configuration of Simulators complicated

Configuration time-consuming and error-prone:
- Can induce side effects into results
- Configuration efforts duplicated

Current workflow when using simulators:
- Familiarize with simulator API
- Model topology and traffic
- Implement topology and traffic in simulator
- Configure simulation environment with external components

⇒ How can problems be mitigated?

Approach: Facilitate easy setup using shareable configuration files
Outline

Challenges of simulative SDN-Application Evaluation

Approach and Implementation

Usage

Conclusion
Appraoches:

- Explicit experiment specification
- Scenario description
- Easy setup using configuration files
Shared Evaluation Environment for SDN

Application ➔ Simulator ➔ Output

Traffic ➔ Topology

Bundle
SEED

Shared Evaluation Environment for SDN

Traffic  Topology

Bundle

Application  Simulator  Output

Shared Evaluation Environment for SDN

Challenges  Approach and Implementation  Usage  Discussion  Conclusion
A. Dittebrandt, M. König, F. Neumeister – Shared Evaluation Environment for SDN-Apps
October, 13th 2017
8/22
Easier reproducibility

Components
- Application(s)
- Scenario
- Simulator
- Parameters

Experiment Description
Scenario-Bundles

(= Traffic + Topology)

- Enable uniform evaluation scenarios
- Fast experiment setup
- Easy sharing & reuse
Format of Scenario-Bundles

Describes complete scenario
- Topology
- Traffic

Properties
- XML-based
- Addressing & grouping of network components
- Process-based traffic generation
- Integration of SDN-components
Simulator-Adapters

Tasks
- Parsing of scenario-bundles
- Connection of SDN-components (via OpenFlow)
- Construction of the topology
- Execution of traffic & events

Implementations
- mininet
- OMNeT++
- ns-3
Unified starting point for experiments:

- Preprocessing of configurations
- Initialization and start of
  - SDN-controller
  - Corresponding SDN-applications
  - Simulation environment
- Connection between components
- Docker Support: Faster setup
Outline:

- Challenges of simulative SDN-Application Evaluation
- Approach and Implementation
- Usage
- Conclusion
Workflow

1. Choose simulator
2. Choose Scenario-Bundle
3. Enter path to own SDN-application
4. Execute SEED
5. Evaluate results

Example usage:
./seed -app pbce -scenario datacenter -simulator ns3
./seed -app ecmp -scenario campus -simulator mininet
Workflow

1. Choose simulator
2. Choose Scenario-Bundle
3. Enter path to own SDN-application
4. Execute SEED
5. Evaluate results

Example usage:

./seed -app pbce -scenario datacenter -simulator ns3
./seed -app ecmp -scenario campus -simulator mininet
Applications:

- **iTAP**: in-network Traffic Analysis Prevention
  - Altering traffic meta-data to randomize communication patterns

- **PBCE**: Port Based Capacity Extensions
  - Migration of flow-rules to neighboring switches

- **ECMP**: Equal Cost Multi Path
  - Multi-path load balancing

Scenarios | Simulators | Applications
--- | --- | ---
Campus | > mn | ECMP
Datacenter | OMNeT++ | ITAP
IXP/ISP | ns-3 | PBCE
Usage Example

Scenarios

- Campus
- Datacenter
- IXP/ISP

Simulators

- > mn
- OMNeT++
- ns-3 Network Simulator

Applications

- ECMP
- ITAP
- PBCE
Usage Example

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Simulators</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Campus</td>
<td>&gt; mn</td>
<td>ECMP</td>
</tr>
<tr>
<td>Datacenter</td>
<td>OMNeT++</td>
<td>ITAP</td>
</tr>
<tr>
<td>IXP/ISP</td>
<td>ns-3</td>
<td>PBCE</td>
</tr>
</tbody>
</table>
# Compatibility

<table>
<thead>
<tr>
<th></th>
<th>ns-3</th>
<th>OMNeT++</th>
<th>Mininet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Controller Support</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>External</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Scenario</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datacenter</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Campus</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ISP &amp; IXP</td>
<td>WIP</td>
<td>WIP</td>
<td>WIP</td>
</tr>
<tr>
<td><strong>SDN-App</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PBCE</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>iTap</td>
<td>WIP</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>ECMP</td>
<td>✓</td>
<td>WIP(^1)</td>
<td>✓</td>
</tr>
</tbody>
</table>

SEED compatibility-matrix

\(^1\) vereinfachte Version
Scope

- Limited by features of simulators
- Limited selection of traffic generators
- Rudimentary node configuration
- OpenFlow only
- Only external controllers
Conclusion:

- Evaluation process, reproducibility & comparability simplified
- SEED-prototype implemented
- Current SEED-version tested
- Example Scenario-Bundles implemented

- Code will be made available as opensource
- Call for Participation: https://git.scc.kit.edu/seed
SEED:
Shared Evaluation Environment for Software-Defined-Network Applications
Further insights

- NS-OF13 extension only supports exactly OpenFlow 1.3 (no backwards-compatibility)
- Implementation of POX 1.3 fork not stable yet
- OMNeT++-extension for OpenFlow 1.3 only implements rudimentary functions
Example Evaluation

- Scenarios
  - Campus
  - Datacenter
  - IXP/ISP

- Simulators
  - OMNeT++
  - ns-3

- Applications
  - ECMP
  - ITAP
  - PBCE

Sources
A. Dittebrandt, M. König, F. Neumeister – Shared Evaluation Environment for SDN-Apps
Results

./seed -app switch -scenario datacenter -simulator mininet
./seed -app pbce -scenario datacenter -simulator mininet

Flowtable-Usage **without** PBCE

Flowtable-Usage **with** PBCE

Sources
A. Dittebrandt, M. König, F. Neumeister – Shared Evaluation Environment for SDN-Apps
October, 13th 2017
Icons: thenounproject.com