



A DevOps Toolkit for Networks

Catalin Meirosu¹, Felician Nemeth², Rebecca Steinert³, Sachin Sharma⁴, Pontus Sköldström⁵, Wolfgang John¹

1 - Ericsson Research; 2 - Budapest University of Technology and Economics; 3 - SICS Swedish ICT; 4 - iMinds; 5 - ACREO Swedish ICT



Motivation

- Research challenges described in draft-unify-nfvrg-devops
- Allow the community to experiment with research results
- Get additional feedback
- Potentially gauge levels of interest in hardening and transitioning some of these tools towards production

The UNIFY project in a nutshell



Major Service Providers:



Research Institutes:



Major Vendors:



Universities:



SMEs:



- Help operators increase the velocity of service introduction
- In WP4, novel observability and verification features usable by both Developers and Operators

The UNIFY DevOps Toolkit

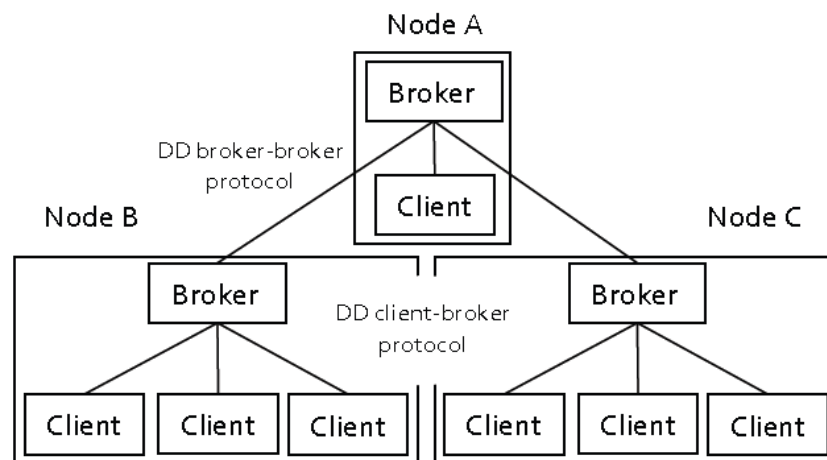
- Is
 - A collection of tools and supporting functions released under individual licenses
 - Provided as-is, with no guarantees
 - Supported by the respective authors' personal best efforts
- Is not
 - Production-ready
 - Optimized for all potential use cases

Components, v1.0, available today

- Support functions
 - Communication bus: DoubleDecker
- Tools
 - Debugging and troubleshooting: EPOXIDE
 - Observability: Rate Monitoring
 - Verification: AutoTPG

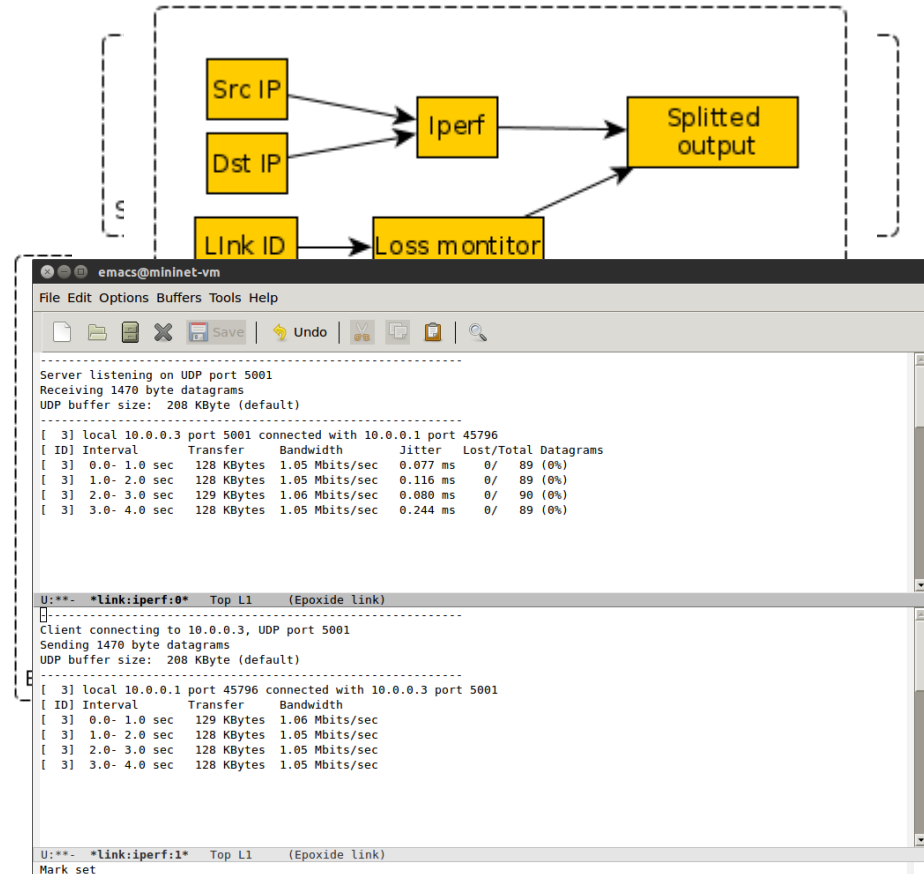
DoubleDecker

- Provides scalable communication services for monitoring functions
 - IP and IPC transport
- Based on ZeroMQ
 - Lighter-weight than RabbitMQ or Kafka
 - No persistence
- Features
 - Client and broker hierarchy
 - Simple routing mechanism
 - Isolation mechanism for multi-tenancy
- <https://github.com/acreo/doubledecker>



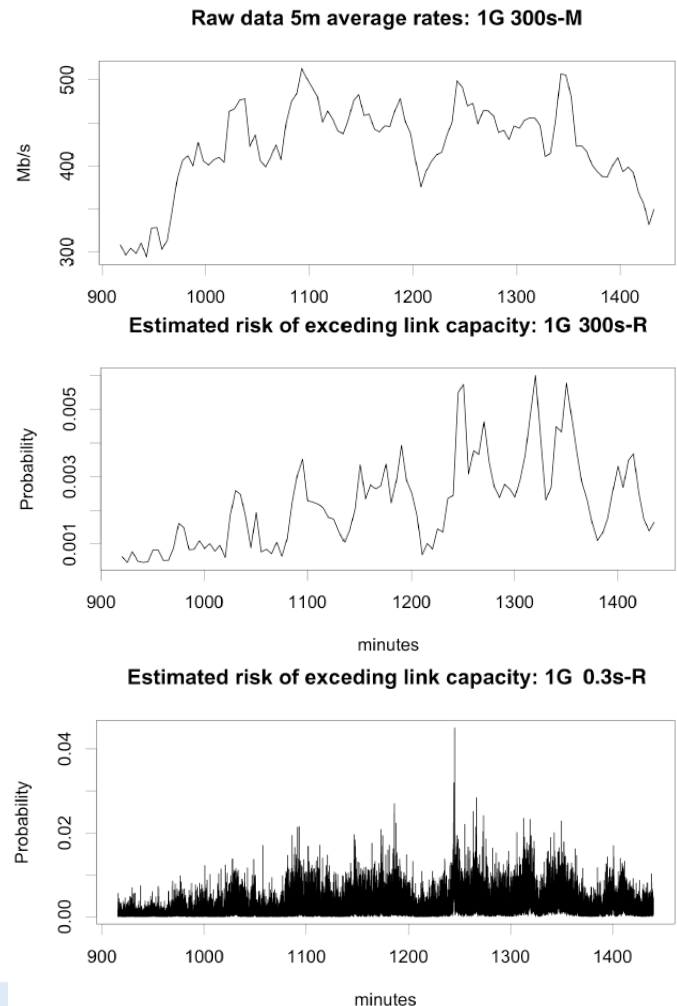
EPOXIDE

- Troubleshooting framework integrating other tools in TroubleShooting Graphs (TSG)
- Based on EMACS
 - Nodes and links of TSG are shown in buffers
 - semantic navigation
 - navigation in a visualized TSG
- Event driven framework
- API for third-party tool developers
- <http://github.com/nemethf/epoxide>



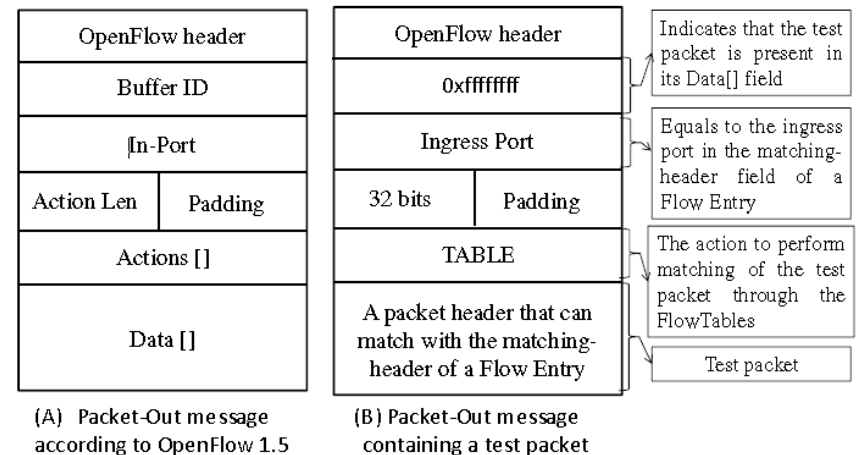
RateMon

- Scalable congestion detector based on reading switch port utilization counters
 - use two counters (first and second statistical moments for link utilization)
 - transmit only the estimate to the controller, instead of high-speed data flow
 - message rate reduction 3000x compared to using raw SNMP counters directly
- <https://github.com/nigsics/ramon.git>



AutoTPG

- Verifies FlowMatch part of OpenFlow descriptors, in particular for aggregated flows
 - Automatically generates test packets for entries under test
 - Binary search for matching error identification
- Identifies
 - Incorrect configuration
 - Bugs in OpenFlow switch implementation
- OpenDaylight application
- <http://users.intec.ugent.be/unify/autoTPG/>



Conclusion

- First version of UNIFY DevOps Toolkit for networks released today
 - individual licenses for each component
 - DoubleDecker: scalable communication bus aimed at transporting monitoring data
 - Rate Monitoring: scalable congestion detector
 - EPOXIDE: framework for integrating troubleshooting tools
 - AutoTPG: verification of aggregate OpenFlow descriptors
- Next steps
 - Looking forward for feedback
 - Add more tools in Spring 2016
 - Verification of Virtual Network Functions
 - Language for describing monitoring intents in conjunction with Network Function Forwarding Graphs

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement no. 619609 - the UNIFY project. The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document