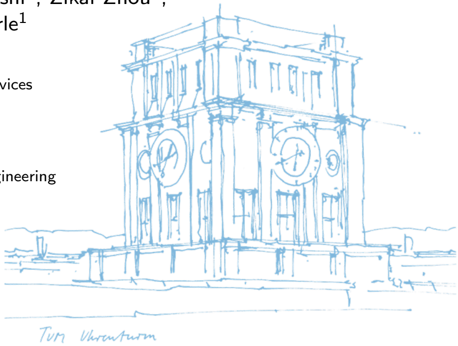
A Framework for Reproducible Data Plane Performance Modeling

Dominik Scholz[1], Hasanin Harkous[2], Sebastian Gallenmüller[1], Henning Stubbe[1], Max Helm[1], Benedikt Jaeger[1], Nemanja Deric[2], Endri Goshi[2], Zikai Zhou[2], Wolfgang Kellerer[2], Georg Carle[1]

[1]Chair of Network Architectures and Services
Department of Informatics

[2]Chair of Communication Networks
Department of Electrical and Computer Engineering

Technical University of Munich



Move to New Programmable Data Planes

P4 Landscape

Wide range of target platforms

- software, e.g. DPDK
- FPGA
- SmartNIC
- ASIC
- "special" platforms, e.g. Raspberry Pi (P4Pi)

Steady stream of new applications

- see P4 workshop(s)
- many papers at this conference
- included at most major conferences
- data center, industrial, mobile, security, . . .

➔ **How will program X perform on target Y?**
➔ **Will scaling match-action tables create a bottleneck?**
➔ **Need to understand performance properties of devices and P4 programs**

Outline

Concept

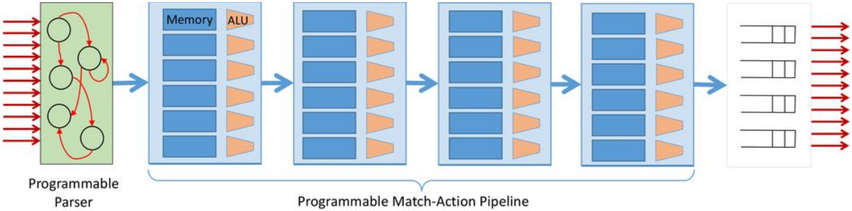


Image from https://bit.ly/3mDpaE9

Based on approach by Dang et al.[1]

- analyze P4 language components individually
- ➔ reduces side-effects
- ➔ detect regressions
- ➔ compare to theoretic performance of underlying algorithm

Scaling components like . . .

- (de)parsed fields
- tables
- table entries
- match width
- . . .

[1] [1] H. T. Dang, H. Wang, T. Jepsen, et al., "Whippersnapper: A P4 language benchmark suite", in Proceedings of the Symposium on SDN Research, SOSR 2017

The Road Towards Reproducible Performance Modeling

Reproducibility requires automation of . . .

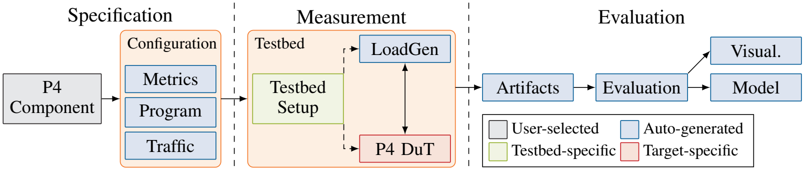
- . . . experiment execution
- . . . evaluation and modeling

Challenges because of broad P4 landscape

- different P4 target platforms ➔ target-specific Device-under-Test (DuT) & metrics
- different P4 architectures ➔ target-specific P4 program
- different testbed environments ➔ testbed-specific setup

➔ **Specification including experiment parameters**

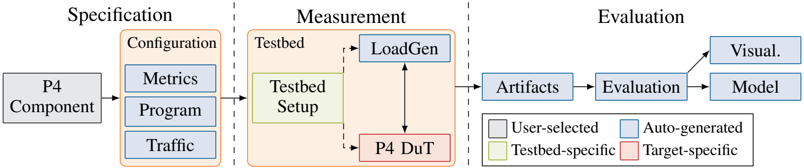
Automated Modeling Framework



Modeling framework overview

Automated Modeling Framework

Specification



Modeling framework overview

Metrics

- packet rate
- latency
- target-specific metrics
- ...

P4 program

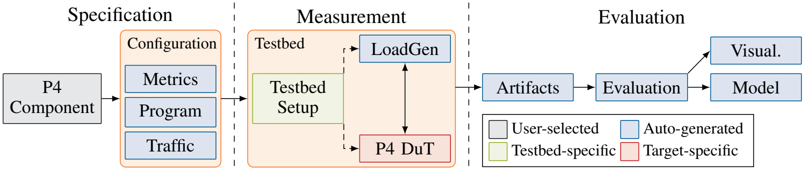
- # parsed headers
- # tables
- # table entries
- ...

Traffic

- packet size
- packet rate
- # headers
- ...

Automated Modeling Framework

Measurement Phase



Modeling framework overview

Minimum physical setup

- two connected nodes
- load generator: MoonGen[2], auto-generated
- DuT: target-specific

Testbed-specific setup

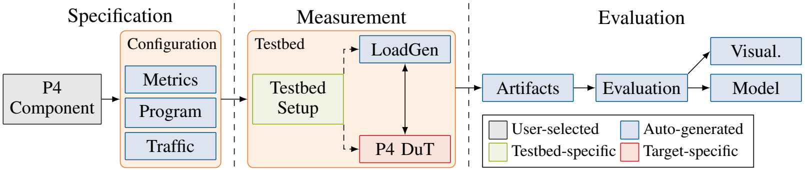
- setup and experiment execution
- plain orchestrating service (pos)[3]

[2] [2] P. Emmerich, S. Gallenmüller, D. Raumer, et al., "Moongen: A scriptable high-speed packet generator", in Proceedings of the 2015 ACM Internet Measurement Conference, IMC 2015

[3] [3] S. Gallenmüller, D. Scholz, H. Stubbe, et al., "The pos Framework: A Methodology and Toolchain for Reproducible Network Experiments", in CoNEXT '21: The 17th International Conference on emerging Networking EXperiments and Technologies, 2021

Automated Modeling Framework

Evaluation Phase



Modeling framework overview

External metrics

- DuT as black-box
- artifacts based on and gathered by MoonGen
- auto-generated ➔ evaluated automatically

Internal metrics

- DuT as white-box
- target-specific artifacts
- requires one-time implementation
- uses same evaluation pipeline

Modeling Approach

We want a model for the DuT.

We have measurement data $g(x) = y$ for measurement domain $x \in G$.

Approach: curve fitting using non-linear least squares for $X \subseteq G$

Solve every function τ from set of possible functions

- polynomials of degree zero to five
- e.g. $\tau(x) = p_1^* x^2 + p_2^* x + p_3^*$
- exponential functions
- logarithmic functions
- inverse of the above

for free parameters $\vec{p^*} \rightarrow \vec{p}$ to match $g(x)$.

How to select best fitting?

Modeling Approach

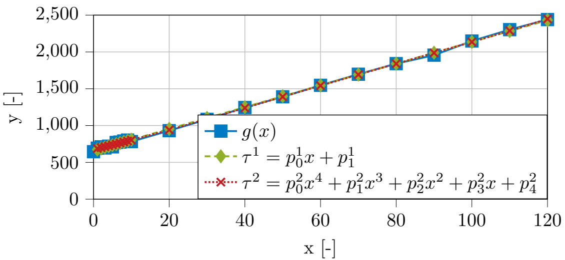
Model Quality Metric

Symmetric Mean Absolute Percentage Error (sMAPE):

$$\eta^{\text{sMAPE}} = \sum_{x \in X} \frac{|\tau(x) - g(x)|}{|g(x)| + |\tau(x)|}$$

Problem: high-degree polynomials will always be preferred

- high-degree parameters close to zero
- smaller error than lower-degree polynomial
- mathematically correct
- rarely represent device behavior



	p_0	p_1	η
τ^1	14.77241	659.44034	1.418%
τ^2	$\lvert p_0 \rvert < 10^{-5}$	-0.00106	1.196%

Two strategies to counteract this behavior

Modeling Approach

Model Quality Metric

Forbid small parameters

- minimum free parameter value γ
- reflects limited measurement accuracy

$$p = \begin{cases} \text{sign}(p) \cdot \gamma, & \text{if } |p| < \gamma \\ p, & \text{otherwise} \end{cases}$$

Variation of Akaike information criterion (AIC)[4]

- idea: penalize complex functions
- assign weight ψ to each function: $\psi = |\vec{p^*}|$
- define margin κ
- pair-wise compare fittings F_1 and F_2
- if difference in error is smaller than κ choose simpler function

$$\tau_{\text{chosen}} = \begin{cases} \tau_1, & |\eta_1 - \eta_2| \leq \kappa, \psi_1 < \psi_2 \\ \tau_2, & |\eta_1 - \eta_2| \leq \kappa, \psi_1 \geq \psi_2 \\ \tau_1, & \eta_1 < \eta_2 \\ \tau_2, & \text{otherwise} \end{cases}$$

➔ **trade-off between accuracy and simplicity of model**

[4] [4] T. S. Chis, "Performance Modelling with Adaptive Hidden Markov Models and Discriminatory Processor Sharing Queues", Ph.D. dissertation, Imperial College London, UK, 2016

Modeling Approach

Resulting Model

One fitting $F = (\tau, \vec{p}, \alpha, \beta, \eta)$ to model the entire measurement domain $X := \{x \in G \mid \alpha \leq x < \beta\}$.

But: complex systems cannot be modeled by a single function

➔ multiple partial fittings

$$\mathcal{F}(x) = \begin{cases} F_1^n(x), & s_0 \leq x < s_1 \\ F_2^n(x), & s_1 \leq x < s_2 \\ \quad\vdots & \\ F_n^n(x), & s_{n-1} \leq x \leq s_n \end{cases}$$

- n individual fittings
- $n+1$ splitting points $\vec{s}$
- $F_i^n = (\tau_i, \vec{p}_i, s_{i-1}, s_i, \eta_i)$
- combined error/rank is weighted sum of individual errors/ranks

➔ combined fitting $\mathcal{F}$ to model entire measurement domain

Modeling Approach

Determining Splitting Points

For up to three partial fittings

- calculate all possible combinations
- $\mathcal{O}(|G|^n)$
- parallelized
- calculation: $< 0.5\,\mathrm{s}$ per individual fitting

More than three partial fittings

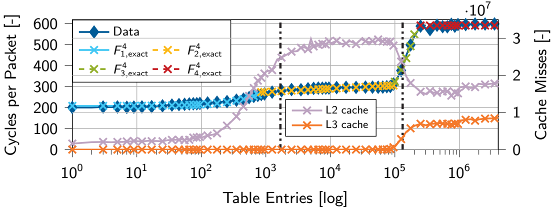
- heuristic: detect performance impacting effects
- → measurement data alters direction of slope
- → second derivative of measurement data
- approximated using local piecewise derivates
- select l indices of absolute extrema

Specification

- DPDK-based t4p4s[5] software target
- scaling # exact match-action table entries
- metric: CPU cycles per packet

Resulting four-split model

- three performance levels modeled
- one "transition" period modeled
- error: 0.730 %
- → can be explained with model for caches misses[6]
- matches theoretic performance of underlying algorithm



[5] [5] P. Vörös, D. Horpácsi, R. Kitlei, et al., "T4P4S: A Target-independent Compiler for Protocol-independent Packet Processors", in IEEE 19th International Conference on High Performance Switching and Routing, HPSR 2018

[6] [6] D. Scholz, H. Stubbe, S. Gallenmüller, et al., "Key Properties of Programmable Data Plane Targets", in Teletraffic Congress (ITC 32), 2020 32nd International

Conclusion

Modeling of individual P4 language components

- reproducibility through automation
- portability through testbed-/target-specific components
- → generated specification
- mathematical model-first approach based on curve fitting
- capable of modeling complex systems

Additional contributions in our paper

- further details on modeling approach
- modeling of recirculation feature on SmartNIC
- discussion of the framework's generalization and customization

Bibliography

[1] H. T. Dang, H. Wang, T. Jepsen, G. J. Brebner, C. Kim, J. Rexford, R. Soulé, and H. Weatherspoon, "Whippersnapper: A P4 language benchmark suite", in Proceedings of the Symposium on SDN Research, SOSR 2017.

[2] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, and G. Carle, "Moongen: A scriptable high-speed packet generator", in Proceedings of the 2015 ACM Internet Measurement Conference, IMC 2015.

[3] S. Gallenmüller, D. Scholz, H. Stubbe, and G. Carle, "The pos Framework: A Methodology and Toolchain for Reproducible Network Experiments", in CoNEXT '21: The 17th International Conference on emerging Networking EXperiments and Technologies, 2021.

[4] T. S. Chis, "Performance Modelling with Adaptive Hidden Markov Models and Discriminatory Processor Sharing Queues", Ph.D. dissertation, Imperial College London, UK, 2016.

[5] P. Vörös, D. Horpácsi, R. Kitlei, D. Leskó, M. Tejfel, and S. Laki, "T4P4S: A Target-independent Compiler for Protocol-independent Packet Processors", in IEEE 19th International Conference on High Performance Switching and Routing, HPSR 2018.

[6] D. Scholz, H. Stubbe, S. Gallenmüller, and G. Carle, "Key Properties of Programmable Data Plane Targets", in Teletraffic Congress (ITC 32), 2020 32nd International.