Report on Networking and Programming Languages 2017

Nikolaj Bjørner
Microsoft Research
nbjorner@microsoft.com

Marco Canini
KAUST
marco@kaust.edu.sa

Nik Sultana
University of Pennsylvania
nsultana@seas.upenn.edu

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The author takes full responsibility for this article's technical content. Comments can be posted through CCR Online.

CCS Concepts

•**Networks** → **Programmable networks;** *Programming interfaces;* Network manageability; •**Software and its engineering** → **Formal software verification;** *Extensible languages;*

Keywords

Programming Languages, Software Defined Networking, Network Verification

ABSTRACT

The third workshop on Networking and Programming Languages, NetPL 2017, was held in conjunction with SIGCOMM 2017. The workshop series attracts invited speakers from academia and industry and a selection of contributed abstracts for short presentations. NetPL brings together researchers from the networking community and researchers from the programming languages and verification communities. The workshop series is a timely forum for exciting trends, technological and scientific advances in the intersection of these communities. We describe some of the highlights from the invited talks through the lens of three trends: Advances in network machine architectures, network programming abstractions, and network verification. NetPL included five invited speakers, four from academia, and one from industry. The program contained six contributed talks out of eight submitted for presentation. The workshop organizers reviewed the abstracts for quality and scope. A total of 42 registrations were received and the attendance occupied the lecture room to the brink. Slides and abstracts from all talks are available from the workshop home page.[1] Videos of the presentations are available in the NetPL YouTube channel.[2]

1. INVITED TALKS

Advances in Network Machine Architectures

The workshop's first keynote, *Programmable Forwarding Planes are Here to Stay* by Nick McKeown, Stanford University, presented an overview of industry and technological trends driving software-defined networking. It concluded with the following highlights:

Chip speed We can now make programmable switch chips as fast as fixed ones.

Chip technology The difference in chip area and power between programmable and fixed function is going away.

Chip complexity There are now too many protocols to correctly hard-code in silicon.

New ideas Beautiful new ideas are owned by the programmer, not the chip designer.

Level playing field Let us create a solid platform, an abstraction layer, upon which more will be built

The case was made that technological advances in chip fabrication allows programmable ASIC to fit in ever shrinking space compared to resources used for lookup tables and packet buffers. Programmability allows to customize networks with a reduced number of features, and thereby complexity, that are suitable to a given deployment. It enables differentiation at the programmable layer and may break silicon lock-in. The P4 programming language[3] is a leading instance of a programming language for packet processing. Two application areas were highlighted, SilkRoad [2] uses P4 for programmable load balancing. Another area enabled by programmability is telemetry and measurement, which was the central theme of the second invited talk.

Network Programming Abstractions

Jennifer Rexford, Princeton University, presented the second invited talk titled *For Good Measure: Programmable Measurement Using Emerging Switches.* As a starting point it highlighted how network measurements are interwoven in the fabric of network engineering. Measurements are used for

Traffic engineering e.g., computing traffic matrices,

Service level agreements e.g., measuring jitter,

Troubleshooting e.g., identifying bottlenecks, and

Security e.g., identifying heavy hitters.

An opportunity presented by programmable data-planes, is that these enable highly customizable, fine-grained and efficient ways to inject measurements in live traffic with minimal overhead. An exemplar of this synergy is NetQRE [4],

[1] http://conferences.sigcomm.org/sigcomm/2017/workshop-netpl.html

[2] https://www.youtube.com/channel/UCqU8E2n4MHthZUVb1xK2nRQ

[3] http://p4.org

which compiles network-wide queries into the packet processing layer. The grander aspiration is for programmability to enable future networks to manage themselves through integration of measurement and control.

Boon Thau Loo, University of Pennsylvania, presented the workshop's third invited talk, *Programming Network Policies by Examples: Platform, Abstraction, and User Studies.* It also presented advances in programming abstractions, but with the perspective of programming language technologies. Advances in program synthesis systems that produce programs from examples have been used as a basis for a network configuration programming system. In the programming from examples area, it is hypothesized that programming (configurations) from examples is easier than using a rigorous formalism or a programming language that incurs a learning curve. To support this hypothesis the talk presented how a user-study was conducted for measuring suitability of the synthesis system.

Network Verification Methodologies

An emerging part of networking conferences are contributions around Network Verification and more broadly Network Design Automation, as exemplified by Boon Thau Loo's presentation on synthesis. As programmability of networking devices is exposed for flexibility there is an urgency to support programming with methodologies, languages, and tools for analysis and synthesis. The past few years has seen a surge in new tools and methodologies aimed at network verification. They borrow techniques, and researchers, from mainstream software analysis, but the domain of networking software presents unique assumptions that can be exploited by network verification tools.

Substantiating this perspective, Mooly Sagiv, Tel Aviv University and VMWare Research Group, presented *Abstractions for Safety of Stateful Networks.* The talk presented a set of recent research results on network verification, including [3]. As an instance of exploiting domain properties, the talk considered verification of stateful middle-boxes. Generally, verification of state machines is intractable, but under the assumption that middle-boxes may revert to initial state through resets, at any time, the talk demonstrated how verification of safety properties can be fully automated.

Wenxuan Zhou, UIUC and Veriflow, presented *Network Verification: From Algorithms to The Real World*, which described the use of verification technologies in Veriflow Systems, Inc. Veriflow addresses the inherent complexity of managing large networks through automated analysis. In common deployments, routes are computed using distributed protocols, such as BGP. Veriflow provides tools for statically checking the resulting state (in contrast to using partial tests, such as traceroute). Zhou further presented how Veriflow's tools, by checking an abstraction of a network, allows to perform change impact analysis on the dataplane. For change impact analysis during deployments, Veriflow developed efficient incremental tools [1], and finally Veriflow offers integration of real-time verification tooling to synthesize network-wide updates. The case was made that an overall capability enabled by commercial Network Verification tools is to deal with networks as one system, in contrast to a network of individual devices, and a single intent-based configuration method. This contrasts managing a collection of decoupled configuration knobs.

There has been a recent surge in startups on network verification. Besides Veriflow, IntentionNet and Forward Networks cater to a market of corporate network operators. Companies in the network verification space provide tools to configure and model network deployments and query the models prior to deployment. Veriflow positions the benefits of Network Verification along four dimensions:

Availability e.g., is the disaster recovery plan going to work?

Resilience e.g., is the network fault tolerant?

Continuous Compliance e.g., does the network meet auditing criteria?

Incident Response e.g., what is the root case of an outage?

Related selling points are made by other industry players, with differentiation on technologies and how they are applied.

It is also exciting times for global cloud providers (Amazon, Azure, among others) that invest in in-house network verification tool chains. We believe that past, and we expect future, editions of NetPL provide an ideal forum for these industry initiatives to share their experiences in a scientific forum.

2. CONTRIBUTED TALKS

In addition to the invited talks, the workshop featured six contributed talks.

Network Protocol Programming in Haskell. K. Yamamoto. This work described the advantages and disadvantages of Haskell being used to write network-related programs, based on the experience of developing a range of such programs (DNS, HTTP, etc) over several years. The speaker described trade-offs from the use of programming platform and compiler support, and the use of this work for standardization efforts.

Dynamic Compilation and Optimization of Packet Processing Programs. G. Rétvári, L. Molnár, G. Enyedi, G. Pongrácz. The idea in this work is to add an optimization phase to run closer to the dataplane (i.e., on the dataplane device) rather than rely entirely on the controller to emit the compiled program to the dataplane. This paper argues that the performance of the dataplane can be improved by knowing the packet-processing behavior in addition to the semantics.

Verifying networks with symbolic execution and temporal logic. M. Popovici, R. Stoenescu, L. Negreanu, C. Raiciu. This work proposes to use a branching-time temporal logic to reason about networks, and thus verify reachability properties about them (concerning the paths followed by packets). It extends prior work on symbolic execution with checking temporal, CTL, properties. It compares to NetPlumber, which is based on regular expressions.

Towards An Auditing Language for Preventing Cascading Failures. E. Zhai, R. Piskac. This proposes a domain-specific language for expressing auditing tasks, which are then analyzed to detect potential failures, and suggest structural changes to the network in order to avoid misconfigurations.

Toward building memory-safe network functions with modest performance overhead. *K. Lee, S. Woo, S. Seo, J. Park, S. Ryu, S. Moon.* In this contribution it was argued that Rust's type system makes it a suitable language for implementing network functions. The talk described an example implementation of a NAT implemented in Rust, and quantity its performance over 40 Gbps links.

Mapping Programmable Network Packet Pipelines to HMC-Enabled FPGAs. *J. Khan, P. Athanas.* This work explores the use of a specific memory device together with FPGA, as a target on which to run P4 programs that take advantage of this memory.

Their two-page abstracts and slides are available from the workshop home page.

3. NETPL: PAST AND FUTURE

NetPL was firstly organized at ECOOP in 2015, by Marco Canini and Robert Soulé and at SIGCOMM 2016, by Marco Canini, Arjun Guha, Robert Soulé, and Nik Sultana. The next edition[4] will be held shortly in conjunction with POPL, on January, Tue 9th, 2018, and will be chaired by Marco Canini, Nate Foster and Todd Millstein.

4. REFERENCES

[1] A. Khurshid, X. Zou, W. Zhou, M. Caesar, and P. B. Godfrey. VeriFlow: Verifying Network-Wide Invariants in Real Time. In *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2013, Lombard, IL, USA, April 2-5, 2013*, pages 15–27, 2013.

[2] R. Miao, H. Zeng, C. Kim, J. Lee, and M. Yu. SilkRoad: Making Stateful Layer-4 Load Balancing Fast and Cheap Using Switching ASICs. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 2017, Los Angeles, CA, USA, August 21-25, 2017*, pages 15–28, 2017.

[3] A. Panda, O. Lahav, K. J. Argyraki, M. Sagiv, and S. Shenker. Verifying Reachability in Networks with Mutable Datapaths. In *14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017, Boston, MA, USA, March 27-29, 2017*, pages 699–718, 2017.

[4] Y. Yuan, D. Lin, A. Mishra, S. Marwaha, R. Alur, and B. T. Loo. Quantitative Network Monitoring with NetQRE. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 2017, Los Angeles, CA, USA, August 21-25, 2017*, pages 99–112, 2017.

[4] https://popl18.sigplan.org/track/netpl-2018