Distributed Software-Defined Networking: ACM PODC 2014 Workshop

Petr Kuznetsov[1] Stefan Schmid[2]

[1] Télécom ParisTech
petr.kuznetsov@telecom-paristech.fr

[2] TU Berlin & T-Labs
stefan.schmid@tu-berlin.de

Abstract

The first workshop on Distributed Software-Defined Networking took place in Paris, France, on the 15th of July, just before the 33rd ACM Symposium on Principles of Distributed Computing. The workshop intended to be a forum to discuss new algorithmic and distributed computing challenges offered by the emerging field of Software Defined Networking (SDN). For the workshop, we invited invited researchers in the fields of distributed computing and networking in order to understand whether distributed implementations of the SDN control plane give rise to new and interesting research questions, in solving which the expertise of the PODC community may be of use.

1 SDN and PODC?

Computer networking currently goes through a phase transition, and the paradigm of Software-Defined Networking (SDN) is discussed intensively, both in the industry and in the academia. In a nutshell, SDN out-sources the control over the network to a logically centralized software, called the *control plane*. The ability of the control plane to "program" the network (the *data plane*) opens new interesting opportunities.

While the perspective of a centralized controller simplifies network management, it comes with the usual drawbacks: single-point of failure, scalability bottleneck, overhead due to indirection, etc. This raises the questions: Do we need a *distributed* control plane for SDN? And if so, what could the "PODC community" contribute?

In order to discuss these questions, we organized the workshop on Distributed Software-Defined Networks (DSDN). The workshop took place in Paris on the sunny day of July 15th, just before PODC, concurrently with four other PODC workshops, and gathered the audience of about 20-40 people (which could be considered as success).

The program consisted of invited and peer-reviewed presentations from researchers working in applied networking, systems, and theory of distributed computing.

2 Program

Below we give a short overview of the talks given in our workshop. Abstracts and slides can be found at `http://www.podc.org/podc2014/dsdn14/`.

2.1 Foundations of SDN

In the opening keynote, Nate Foster (Cornell) gave an overview of programming abstractions for SDN. The talk consisted of three parts.

...

2.2 Consistent Range Classification with OpenFlow

Yehuda Afek (TAU) presented his recent work with Anat Bremler-Barr and Liron Schiff [1].

...

2.3 SDN-Based Private Interconnection

...

2.4 Software Transactional Networking: A Robust and Distributed SDN Control Plane

Marco Canini (UCL) presented the concept of software transactional networking (STN), a control-plane abstraction used for consistent composition of concurrent policies [2, 3]. The abstraction assumes a set of control applications that concurrently apply modifications (or updates) of the *network policy*, i.e., the set of rules that stipulate how the traffic should be processed at the data plane. Given that the policy updates coming from different control applications may conflict with each other, the STN framework offers to the applications a *transactional* interface with *all-or-nothing* semantics. The talk hinted on the formal definition of the abstraction of consistent policy composition (CPC), and sketched designs of its resilient implementations.

2.5 Declarative, Distributed Configuration

Can the challenges motivating the use of SDN be addressed using existing hardware and protocols. Sanjai Narain (Applied Communication Sciences) summarized the experience collected at his company in using the Assured and Dynamic Configuration(ADC) system. In their approach, network functionality is expressed as a set of constraints on configuration variables. SAT or SMT solvers are used to convert these constraints into values of configuration variables. Of course, in specific scenarios, proprietary solutions may be more efficient and easier to deploy than generic ones (e.g., based on the SDN framework), and Sanjai's talk questioned the very motivation behind migrating to SDN.

2.6 Managing the Network with Merlin

...

2.7 Managing Dynamic Networks: Distributed or Centralized Control

In his somewhat provocative talk, Roger Wattenhofer (ETHZ) considers the following question. If we take the extreme case of a network managed (e.g., using the SDN approach) by a (fault-tolerant, performant, etc.) central controller, would this imply that distributed algorithms are no longer needed. The answer (not very surprisingly) is no: we still have to deal with the problem

of dynamicity and failures on the data plane, as well as the fact that the data plane is inherently geographically distributed and cannot be manipulated in the atomic manner. In short, we still have to deal with *consistency* of network control, and here we can benefit from distributed computing which is essentially all about consistency. In his talk, Roger overviewed several natural network consistency criteria (such as *loop-freedom* or *per-packet consistency*) and sketched several impossibility results and complexity bounds of achieving these criteria in a few different network models.

3 Distributed SDN: New? Interesting?

A recently published detailed survey on the work that has been in SDN-related research so far [4] claims that "the myth that logical centralization implied a physically centralized controller" has been addressed and, as a result, "SDN ideas have matured and evolved from an academic exercise to a commercial success". Interestingly, among 407 articles cited in the survey, we do not find a single paper that appeared in a distributed-computing theory venue (such as PODC or DISC). We think, based on our personal research experience, this is not because SDN does not give rise new and interesting research questions related to distributed computing. Maybe it is just lack of curiosity or insights on recent advances in networking?

The workshop shows that the situation is slowly changing: a few research groups are looking at the distributed aspects of SDN now. It seems, however, that progress here is still rather modest and typically boils down to solving "conventional" algorithmic problems, using SDN only as a motivating application. We believe nevertheless that there are deeper distributed challenges coming directly from SDN.

References

[1] Y. Afek, A. Bremler-Barr, and L. Schiff. Cross-entrance consistent range classifier with openflow. In *ONS*, 2014.

[2] M. Canini, P. Kuznetsov, D. Levin, and S. Schmid. Software transactional networking: concurrent and consistent policy composition. In *HotSDN*, pages 1–6, 2013.

[3] M. Canini, P. Kuznetsov, D. Levin, and S. Schmid. A distributed sdn control plane for consistent policy updates. Technical report, arXiv TR 1305.7429, 2014.

[4] D. Kreutz, F. M. V. Ramos, P. Verssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-defined networking: A comprehensive survey. Technical report, arXiv TR 1406.0440, 2014.