Towards meticulous data plane monitoring

Apoorv Shukla†  Said Jawad Saidi†  Stefan Schmid†  Marco Canini‡  Anja Feldmann†
†TU Berlin  †‡Aalborg University  ‡KAUST

I. Introduction
Software-defined networks (SDNs) are built on the promise of consistency between control and data plane. Data plane consists of software or hardware and is subject to failures e.g., hardware or software bugs. Sometimes, these failures are outside the knowledge of the control plane consisting of network operating system or the SDN controller. The current state-of-the-art tools either check the network statically from the control plane or use tagging or active probe generation to check the data plane behaviour. There is a spectrum of control plane mechanisms [1], [2], [3], [4] which check for some or all of the network-wide invariants like reachability, forwarding loops, waypoint enforcement and slice isolation. While most of the control plane mechanisms cannot model ECMP or NAT, all of them, however, are unaware of the actual behaviour on the data plane and thus, assume that control plane has a consistent view. There is a panoply of data plane mechanisms involving tagging [5], [6] or active probe generation [7], [8] which either check the path or the installed flow rules to monitor the data plane. Moreover, ping and traceroute continue to be the only tools available for network debugging to the network administrators in practice. Bugs in flow table match-action logic or switch hardware can manifest in various ways which are hard to detect and thus, to localize by any of the existing approaches. Moreover, tagging comes with its own limitations. There cannot be available and sufficient space for placing tags for a big scale scenario. Secondly, tags grow with paths and a limited space field header is not sufficient in a scalable scenario. Thirdly, specific flow rules should be installed in order to take actions on the tagged packets. TCAM is a scarce storage resource and therefore, installing additional flow rules may make TCAM space insufficient for adding further forwarding rules.

On the control plane front, we conducted experiments with the existing control plane mechanisms [1], [2], [3]. HSA [1] and NetPlumber [2] carry out set operations like union, difference, complement and intersection on wildcards representing packet headers. The set operations are efficient on Boolean expression but not on wildcards which make HSA and NetPlumber slow. Binary Decision Diagrams (BDDs) [10] are data structures for boolean expressions and have a much faster support for set operations. APVerifier [3] uses BDD-based header space analysis. An implementation based on APVerifier allows
fast control plane-level calculations to figure out packet header spaces reachable at any point in the topology under consideration. APVerifier has two models one with BDD and the other with Atomic Predicates based on BDD. We carried out our experiments in an ECMP simulation where each node had two links to the next node. At the first node, if the first bit had zero in first bit position, the packet was sent on first link to second node and if there was one in the first bit, the packet was sent to the other link to second node. On the second node, the second bit of the packet was checked for forwarding in the similar fashion. We continued upto 50 nodes. Figure 1 shows the execution time of HSA, NetPlumber and the two variations of APVerifier. The experiment carried out on OpenFlow-based OpenVSwitches clearly shows that in our ECMP simulation environment, our BDD-based implementation of APVerifier performed better.

III. Research Directions

For an efficient data plane monitoring mechanism, we need to use our key insight to develop a new header field dedicated for tagging. In order to ensure that the data plane monitoring is rigorous, we would like to field dedicated for tagging. In order to ensure that the network is meticulously covered by the control plane, we need a rigorous data plane verification mechanism that not only checks paths but also the rules matched by the packet on its way. This way we ensure that the network is meticulously covered by the monitoring approach.

IV. Summary

Network monitoring is a task of utmost importance as bugs or misconfigurations could seriously impact the network leading to loss of revenues. In order to make sure that the data plane is consistent with the control plane, we need a rigorous data plane verification mechanism that not only checks paths but also the rules matched by the packet on its way. This way we ensure that the network is rigorously covered by the monitoring approach.

Acknowledgement This work was supported by Leibniz Prize project funds of DFG - German Research Foundation: Gottfried Wilhelm Leibniz-Preis 2011 (FKZ FE 570/4-1).

References