VeriFlow: Verifying Network-Wide Invariants in Real Time

Ahmed Khurshid*, Xuan Zou*, Wenxuan Zhou*, Matthew Caesar, P. Brighten Godfrey
University of Illinois at Urbana-Champaign
{*khurshi1, xuanzou2, wzhou10, caesar, pbg*}*@illinois.edu*
*Student

Abstract

Networks are complex and prone to bugs. Existing tools that check network configuration files and the data-plane state operate offline at timescales of seconds to hours, and cannot detect or prevent bugs as they arise.

Is it possible to *check network-wide invariants in real time*, as the network state evolves? The key challenge here is to achieve extremely low latency during the checks so that network performance is not affected. In this poster, we present a design, VeriFlow, which suggests that this goal is achievable. VeriFlow is a layer between a software-defined networking controller and network devices that checks for network-wide invariant violations dynamically as each forwarding rule is inserted, modified or deleted. VeriFlow supports analysis over multiple header fields, and an API to support invariant-checking beyond simple reachability analysis. Based on a prototype implementation integrated with the NOX OpenFlow controller, and driven by a Mininet OpenFlow network and Route Views trace data, we find that VeriFlow can perform rigorous checking within hundreds of microseconds per rule insertion or deletion.

In order to ensure real-time response, VeriFlow introduces novel incremental algorithms. First, we slice the network into a set of *equivalence classes* (ECs) of packets based on the new rule and the existing rules that overlap with the new rule. Packets belonging to an EC experience the same forwarding actions throughout the network. We find the set of ECs whose operation could be altered by a rule, and verify network invariants only within those classes. Second, VeriFlow builds individual *forwarding graphs* for each of these ECs using the current network state. Third, VeriFlow traverses these graphs to *query* the status of one or more invariants.

Our implementation of VeriFlow supports checking of both OpenFlow [5] version 1.1.0 and IP forwarding rules, with the exception that the current implementation does not support actions that modify packet headers. We microbenchmarked VeriFlow using a stream of updates from a simulated IP network, constructed with Rocketfuel [1] topology data and real BGP traces collected from Route Views [2]. We find that VeriFlow is able to verify network-wide invariants within hundreds of microseconds as new rules are introduced into the network (Figure 1).

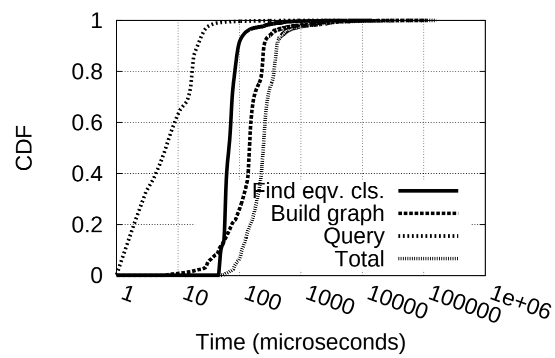

Figure 1: Microbenchmarks using a realistic trace of forwarding table updates show VeriFlow's verification time remains below 1 millisecond for 97.8% of the updates.

This poster summarizes published results in HotSDN 2012 [3] and an upcoming paper in NSDI 2013 [4]. A demo will also be set up with the poster.

References

[1] Rocketfuel: An ISP topology mapping engine. http://www.cs.washington.edu/research/networking/rocketfuel/.

[2] University of Oregon Route Views Project. http://www.routeviews.org/.

[3] KHURSHID, A., ZHOU, W., CAESAR, M., AND GODFREY, P. B. VeriFlow: Verifying network-wide invariants in real time. In *HotSDN* (2012).

[4] KHURSHID, A., ZOU, X., ZHOU, W., CAESAR, M., AND GODFREY, P. B. VeriFlow: Verifying network-wide invariants in real time. In *NSDI* (2013).

[5] MCKEOWN, N., ANDERSON, T., BALAKRISHNAN, H., PARULKAR, G., PETERSON, L., REXFORD, J., AND SHENKER, S. OpenFlow: Enabling innovation in campus networks. In *SIGCOMM CCR* (2008).