Open Network Emulator (ONE), Production grade emulation at scale

Haseeb Niaz, Sreenivas Addagatla, Neha Raje
Yibo Zhu, Jitendra Padhye

June 15, 2018

1 Introduction

In this demo, we present Open Network Emulator (ONE), a network emulator system in daily use at Microsoft. We are planning to open-source it because we believe ONE would benefit not only big enterprises like Microsoft, but also the academia and research community.

Network reliability is critical for large clouds and online service providers like Microsoft. Our network is large, heterogeneous, complex and undergoes constant churns. In such an environment, even small issues triggered by device failures, buggy device software, configuration errors, unproven management tools and unavoidable human errors can quickly cause large outages. Therefore, the ability to validate the impact of every planned change in a realistic setting, before the change is deployed in production, is crucial to maintaining and improving the reliability of our network.

In-situ network verification technologies (e.g. VeriFlow [1]) are not sufficient, since they cannot, by definition, prevent a problem before it occurs. Logical network verification (e.g. Batfish [2]) is not sufficient, since it lacks realism, and it misses errors that stem from interaction between configuration and the control plane software. Indeed, based on the root cause analysis of the incidences in the past two years, only one third of incidences could have been caught by logical verification. Small-scale testbeds and emulations (e.g. MiniNet, Emulan) are also not sufficient, since they do not support the scale and heterogeneity that we need.

To this end, we have built a high-fidelity, cloud-scale network emulator, called Open Networking Emulator (ONE) [3]. ONE is capable of faithfully emulating a network consisting of thousands of heterogeneous devices. The key idea behind ONE is to run a virtualized version of each device in a sandbox (container or VM), and inter-connect the sandboxes with virtual network links according to real network topologies. We then inject real configurations into the virtual devices, to form an emulation that provides faithful replication of the network control plane, and a reasonable approximation of the network data plane. Most importantly, ONE is completely transparent from a network engineer's perspective – the engineer interacts with the devices the same way she would with a normal production hardware device.

ONE has been in daily use in Azure for past 12 months. During this time, we have spent millions of core-hours on ONE emulations – and caught hundreds of bugs in proposed changes, potentially preventing major outages.

We are planning to open-source ONE. Apart from validating network changes, ONE has many other uses. Router vendors can use ONE to test new control plane software at scale. Academic instructors can use ONE to create hands-on, realistic, large-scale lab exercises for their students. Doing this on real hardware would be prohibitively expensive. ONE can also be used for deeper, professional training on specific devices (e.g. Cisco certification exams). Finally, academic researchers can validate their ideas on a high-fidelity and large-scale emulated environment that was not available before. In fact, multiple hardware vendors, companies that operate their own networks, and academic researchers have been in contact with us on using ONE.

2 Demo details

In the demo we will demonstrate the various ways in which ONE is used within Microsoft.

- We will demonstrate how ONE is used to emulate a data center network using the exact same topology, switch OS and configurations as what is seen in the production data center.
- We will showcase a copy of Microsoft's WAN network in ONE running as a perpetual emulation. This copy is used to test out all changes before they get applied to the production network.
- We will also demonstrate how ONE is used to test out inter data center changes by connecting *on demand* emulations of data centers with the perpetual emulation of the WAN network.
- We will also demonstrate how ONE is used in conjunction with hardware switches to create a mixed-field environment for the testing of hardware specific functionalities.
- We will also be going over how instructors can leverage ONE in the classroom by demonstrating some common routing problems like the *Stable Path Problem*[4] and *Count to infinity problem* on ONE.

References

[1] A. Khurshid, W. Zhou, M. Caesar, and P. Godfrey, "Veriflow: Verifying network-wide invariants in real time," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 49–54.

[2] A. Fogel, S. Fung, L. Pedrosa, M. Walraed-Sullivan, R. Govindan, R. Mahajan, and T. D. Millstein, "A general approach to network configuration analysis." in *NSDI*, 2015, pp. 469–483.

[3] H. H. Liu, Y. Zhu, J. Padhye, J. Cao, S. Tallapragada, N. P. Lopes, A. Rybalchenko, G. Lu, and L. Yuan, "Crystalnet: Faithfully emulating large production networks," in *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 2017, pp. 599–613.

[4] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Transactions on Networking (ToN)*, vol. 10, no. 2, pp. 232–243, 2002.