Try Before you Buy: SDN Emulation with (Real) Interdomain Routing

Brandon Schlinker[1], Kyriakos Zarifis[1], Italo Cunha[2], Nick Feamster[3], Ethan Katz-Bassett[1], and Minlan Yu[1]

[1]Department of Computer Science, University of Southern California
[2]Department of Computer Science, Universidade Federal de Minas Gerais
[3]School of Computer Science, Georgia Institute of Technology

Abstract

Adoption of Software Defined Networking (SDN) remains largely confined to data center networks, where network architects frequently have the luxury of designing and deploying within a tightly controlled greenfield environment. In comparison, adoption of SDN technologies within existing ISP and enterprise networks depends on the network operator's ability to evaluate SDN's impact and define a transition plan with confidence. Unfortunately, existing tools for emulating SDN networks do not support the evaluation of interactions with components found in today's networks, such as intradomain routing policies and interdomain interactions with other autonomous systems. Within this work, we explain how our testbed provides network operators with comprehensive, 1:1 emulation of an existing network, which can then be used to experiment with SDN and evaluate its impact. We believe that our testbed will provide operators with a better understanding of how SDN *fits* into their existing networks, thus making SDN technologies more accessible to the broader community.

1 Motivation

Software Defined Networking (SDN) promises to change the networking landscape by shifting control of network infrastructure from individual devices to a logically centralized control plane. Yet despite SDN's advantages, adoption remains largely confined to data center networks, where network architects frequently have the luxury of designing and deploying within tightly controlled greenfield environments. In comparison, enterprise and ISP networks are often the result of piecemeal implementations performed over many years, creating a complex patchwork of infrastructure, policies, and routing domains. In addition to this internal chaos, these networks must rely on the Border Gateway Protocol (BGP) to manage their complex peering arrangements with other autonomous systems (AS). The impact of these interdomain relationships extends beyond the edge of the network, as the routes exchanged with peer networks define internal traffic patterns. All of this added complexity makes it significantly more difficult for network operators to adopt SDN technologies, let alone understand how adoption will impact their networks.

We believe that network operators will be better positioned to adopt SDN technologies if they can gain a holistic understanding of how such technologies fit into their networks. However, while tools such as Mininet [1] can emulate a complex, pure SDN network topology, they are unable to emulate interactions with the legacy routing protocols found in today's large networks or a network's interactions with other autonomous systems on the Internet. In particular, Mininet is by default unable to simultaneously execute multiple instances of a routing engine, such as Quagga, making emulation of a large legacy network impossible. As a result, a network operator is unable to use Mininet alone to perform a comprehensive emulation and evaluation of their existing network architecture.

In addition to helping network operators adopt SDN, the need for such comprehensive evaluation will only continue to grow as SDN begins to be used at the network edge to facilitate traffic engineering objectives and centralize BGP interactions [3]. Supporting evaluations which seek to understand the impact of these changes requires an emulation environment capable of supporting legacy routing protocols. But just providing support for these protocols is insufficient, as a comprehensive evaluation must also take into account the unknown polices and unpredictable failures of the Internet. Unfortunately, no level of emulation is capable of accurately modeling these properties, making emulation alone ineffective for evaluating the impact of interactions with other ASes [2].

2 Comprehensive Emulation with Real Connectivity

From our survey of existing emulation tools, it is clear that evaluating the integration of SDN into an existing network architecture requires more than what today's tools are capable of. In recognition of this limitation, we have developed

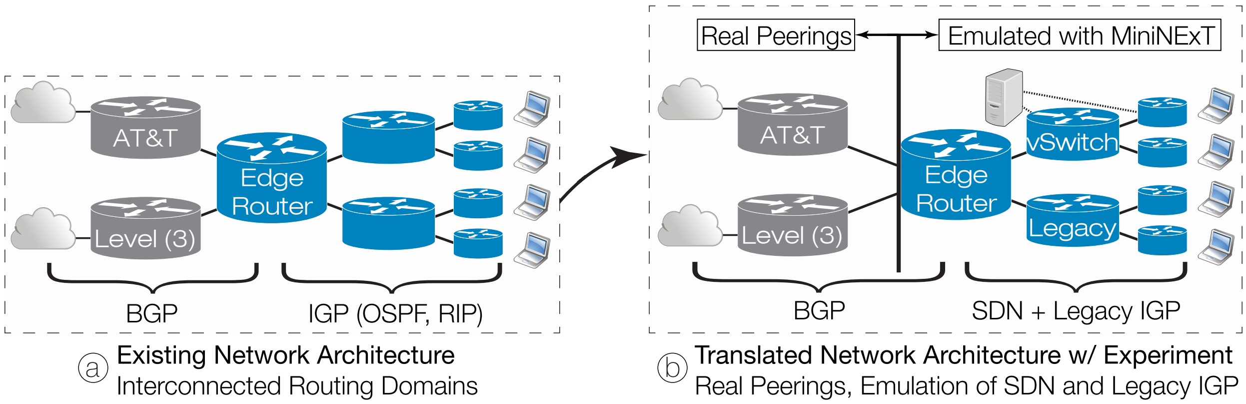

Figure 1: Existing network and translated network (using testbed) with partial adoption of SDN technologies

a testbed that combines comprehensive emulation of internal routing domains with the ability to exchange routes and traffic with other autonomous systems on the *real* Internet.

At a minimum, translating an existing large network into an emulation environment requires that the emulator support existing Interior Gateway Protocols (IGPs) and the Border Gateway Protocol. To this end, we built MiniNExT, an extended version of the Mininet emulator. MiniNExT supports the emulation of networks that extend beyond the pure SDN scope by making it possible for legacy routing engines (such as Quagga) to run within a MiniNExT container. As a result, MiniNExT can emulate hybrid networks that involve both legacy IGP protocols and SDN technologies. Emulating such a network with MiniNExT also uses significantly less resources in comparison to using virtual machines, enabling a single laptop to emulate a complex network with hundreds, if not thousands of legacy routing engines.

In addition to supporting legacy routing protocols, MiniNExT provides emulated networks with *real* Internet connectivity, including the ability to exchange routes and traffic with neighboring domains. It can achieve this connectivity by either connecting to a network operator's *real* network at existing points-of-presence (PoPs), or by connecting to *Transit Portal* [5], a BGP multiplexing service available to the networking community. When MiniNExT connects to a network operator's existing PoPs for connectivity, we use multiplexing (similar to FlowVisor [4]) and other restrictions to isolate all traffic and routing announcements from emulated networks, ensuring that the operator's existing *real* network is not negatively impacted by an experiment.

With MiniNExT, a network operator can clone their *real* network architecture into an emulated environment and then evaluate the impact of changes to their existing architecture. We believe that MiniNExT will provide network operators with greater insight into how SDN will impact their network and enable them to define transition plans with confidence, while providing researchers with an invaluable tool for network experimentation.

3 A Closer Look: Translating an Existing Network

A network operator can use MiniNExT to *clone* the architecture of an existing network, including the network's substrate, its routing complexities, and its peerings with other ASes. The resulting emulated network is shown within Figure 1. To enable the emulated network to communicate with other ASes, the operator has allocated resources from their existing PoPs, such as dedicated IP prefix space. The emulated components within the testbed use these resources to connect via the operator's existing PoPs to the larger Internet, enabling the exchange of routes and traffic. This example also shows that the emulated network has an ongoing experiment, with a portion of the network modified to use SDN. With the help of MiniNExT, operators will be able to better understand the impact of SDN on their existing networks and develop transition plans with confidence, improving the adoption rate of SDN.

References

[1] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown. Reproducible network experiments using container-based emulation. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 253–264. ACM, 2012.

[2] R. V. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. In search of the elusive ground truth: the Internet's AS-level connectivity structure. In *ACM SIGMETRICS Performance Evaluation Review*, volume 36, pages 217–228. ACM, 2008.

[3] Open Networking Foundation. Migration use cases and methods, 2014.

[4] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar. Can the production network be the testbed? In *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, OSDI'10, pages 1–6, Berkeley, CA, USA, 2010. USENIX Association.

[5] V. Valancius, H. Kim, and N. Feamster. Transit portal: BGP connectivity as a service. *ACM SIGCOMM Computer Communication Review*, 40(4):463–464, 2010.