Incremental SDN Deployment in Enterprise Networks

Dan Levin* Marco Canini Stefan Schmid Anja Feldmann
TU Berlin / T-Labs
<first name>@net.t-labs.tu-berlin.de

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Network operating systems

Keywords

Software Defined Network, Incremental Deployment

1. INTRODUCTION

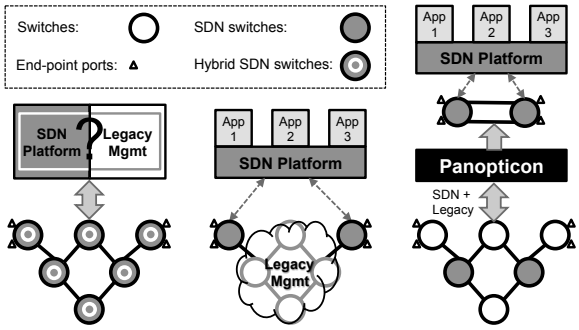
Mid to large enterprise campus networks must operate reliably and provide high-performance connectivity while enforcing organizational policy. They must also provide isolation across complex boundaries, yet remain easy to manage. All the while, operational and capital costs must be kept low. Software Defined Networking (SDN) has the potential to provide a principled solution to these complex operational challenges. However, most existing work to leverage SDN (*e.g.*, [2,4,7]) has so far assumed a full SDN deployment.

Unlike datacenter networks, enterprise network upgrade begins not with a green field, but with the existing deployment and is typically a staged process. Budgets are constrained, and only a part of the network can be upgraded at a time—SDN deployment in the enterprise is no exception. The realities of network upgrade and the operational challenges facing existing networks lead us to question: (i) What are the benefits of upgrading to a partial SDN deployment? (ii) How do the benefits of principled network orchestration depend on the location of SDN switches? (iii) Given budget constraints, what subset of legacy switches or routers should be SDN upgraded to maximize benefits?

To answer these, in this demo we present **Panopticon**, an architecture and methodology for aiding operators in planning and operating networks that combine legacy switches and routers and SDN switches. We call such networks *transitional networks.* We show how Panopticon exposes an abstraction of a fully-deployed SDN in a partially upgraded transitional network, where the SDN benefits extend potentially over the entire network. Panopticon overcomes many of the limitations of current approaches for transitional SDN deployments, which we now briefly review.

*Dan Levin is the only student author.

SIGCOMM'13, August 12–16, 2013, Hong Kong, China.
ACM 978-1-4503-2056-6/13/08.



(a) Dual-stack (b) Access edge (c) Panopticon

Figure 1: Current transitional network approaches vs. Panopticon: (a) Dual-stack ignores legacy and SDN integration. (b) Full edge SDN deployment enables end-to-end control. (c) Panopticon partially-deployed SDN acts like a full SDN deployment.

2. TRANSITIONAL SDN DEPLOYMENT

The first SDN deployment approach (Figure 1a) partitions the flow space into several disjoint slices and assigns each slice to either SDN or legacy processing [6]. Individual traffic flows of interest may be explicitly selected for SDN processing. This mode's prime limitation is that it is essentially a dual-stack approach (as with IPv6 + IPv4) rather than a means to integrate legacy hardware and expose the resulting transitional network as SDN. Further, this approach necessitates a contiguous deployment of hybrid programmable switches capable of processing packets according to both legacy and SDN mechanisms.

The second approach (Figure 1b) involves deploying SDN at the network access edge [3]. This mode has the benefit of enabling full control over the access policy and the introduction of new network functionality at the edge, *e.g.*, datacenter network virtualization [1]. Unlike a datacenter environment where the network edge may terminate at the VM hypervisor, the enterprise network edge terminates at an access switch. In an enterprise network, this approach thus involves upgrading thousands of access switches and incurs a high cost. SDN deployment limited to the edge additionally impairs the ability to control forwarding decisions within the network core (*e.g.*, load balancing, middlebox traversal).

3. THE PANOPTICON APPROACH

In contrast to existing approaches, **Panopticon** (Figure 1c) fundamentally integrates legacy and SDN switches and exposes to the control platform an abstraction of the physical network as a logical SDN. Our main insight is that *the key benefits of the SDN*

paradigm to enterprise networks can be realized for every source-destination path that includes at least one SDN switch. Thus, we do not mandate a full SDN deployment—a relatively small subset of all switches may suffice. Each path which traverses even just one SDN switch, can be used to realize a programmatic, logically-centralized interface for orchestrating *e.g.*, the network access control policy. Moreover, traffic which traverses two or more SDN switches may be controlled at even finer levels of granularity enabling richer forwarding decisions *e.g.*, for load balancing.

Based on this insight, we have developed a **cost-aware optimization tool** for the network operator to determine the topological location of the partial SDN deployment based on their objectives (*e.g.*, CAPEX or forwarding efficiency). Second, we have designed the *Panopticon* architecture for transitional networks which provably **guarantees** that traffic destined to operator-selected end-points passes through at least one SDN switch. Just as enterprise networks regularly divert traffic (*e.g.*, one VLAN to reach another on the *same* switch must traverse a gateway), Panopticon explicitly leverages waypoints to control traffic and thus realize the SDN abstraction.

Analogous to its architectural namesake, Panopticon isolates end-hosts in the legacy network using VLANs (in what we term Solitary Confinement Trees or SCT) and restricts their traffic to traverse strategically upgraded SDN switches. To realize this behavior, Panopticon must overcome the challenges of (i) the need to maintain compatibility with legacy switches and protocols, and (ii) scalability issues with VLAN and flow table state. Panopticon maintains legacy compatibility by relying only on mechanisms which are ubiquitously available on enterprise-grade switches. Through rigorous design, Panopticon minds VLAN and flow table constraints, allowing us to scalably isolate every network end-point subject to SDN waypoint-enforcement.

4. FEASIBILITY

We evaluate the feasibility of our approach with a prototype implementation, as well as through simulation on real enterprise campus network topologies entailing over 1500 switches and routers, using real enterprise network traffic traces. Our results suggest that with only a handful ($< 0.6\%$) of upgraded switches, it becomes possible to operate most ($> 80\%$) of an enterprise network as a single SDN while meeting key VLAN and flow table resource constraints. The primary scaling factor of our system is the number of flow table entries–which is problematic when many wildcard matching rules are needed. While Panopticon exposes a logical SDN abstraction from the underlying partial-SDN deployment, the SDN global network view is reduced to the set of upgraded switches. We now focus on what this means for the SDN programming abstraction:

Panopticon SDN vs. full SDN. As opposed to conventional links in a fully-deployed SDN, links in Panopticon are pseudo-wires composed of legacy switches and links that run STP. Accordingly, the SDN controller must take into account the behaviors of STP within each pseudo-wire. For example, an STP reconvergence in an SCT can create the impression that a pseudo-wire "jumps" from one SDN switch to another–however, such behavior can be accounted for by the SDN control platform.

Hiding the partial deployment from the app. In Panopticon, each SDN-controlled network end-point is not necessarily directly attached to a SDN switch port, but rather to a legacy switch that merely ensures all traffic toward any other destination traverses *some* SDN switch. Consequently, the SDN control platform may present or hide this information from the application to conceal the nature of the partial-deployment as needed. For example, if a control application wants to see the first packet of a flow to know from where the packet originated, the control platform can conceal the pseudo-wire nature of the legacy network, such that—to the application—the packet appears to arrive from an end-point and not the physically neighboring legacy switch.

Which SDN applications are possible in Panopticon? Panopticon enables the expression of any end-to-end policy, as though the network were one big, virtual switch. End-to-end policies include access control and application load balancing. Routing and path-level policy, *e.g.*, traffic engineering can be expressed too, however the fidelity of the global network view (and path diversity) presented to the control logic is reduced to the fidelity of the logical SDN. As more of the network is upgraded to support SDN, more fine-grained path-level policy can be expressed.

Why fully-deploy SDN in the enterprise? Perhaps many enterprise networks do not need to fully deploy SDN. Our Panopticon evaluation suggests that partial deployment may in-fact be the right long-term approach for some enterprise networks based on the prevailing budget limitations and resource constraints.

In summary, Panopticon enables us to expose an interface for operating a transitional network as if it were a fully deployed SDN, and reap the benefits of partial SDN deployment for most of the network, not just the part that is upgraded. Our technical report [5] provides a detailed description and evaluation of Panopticon.

5. REFERENCES

[1] Nicira Network Virtualization Platform. `http://nicira.com/en/network-virtualization-platform`.

[2] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker. Ethane: Taking control of the enterprise. In *SIGCOMM*, 2007.

[3] M. Casado, T. Koponen, S. Shenker, and A. Tootoonchian. Fabric: a retrospective on evolving SDN. In *HotSDN*, 2012.

[4] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker. Frenetic: A network programming language. In *ICFP*, 2011.

[5] D. Levin, M. Canini, S. Schmid, and A. Feldmann. Panopticon: Reaping the Benefits of Partial SDN Deployment in Enterprise Networks. Technical report, TU Berlin / T-Labs, May 2013.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: enabling innovation in campus networks. *SIGCOMM CCR*, 38(2), 2008.

[7] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, and D. Walker. Abstractions for network update. In *SIGCOMM*, 2012.