Panopticon: Incremental Deployment of Software-Defined Networking

Marco Canini[◇] Anja Feldmann[†] Dan Levin[†] Fabian Schaffert[†] Stefan Schmid[†•]
[◇] Université catholique de Louvain [†] TU Berlin [•] Telekom Innovation Labs

ABSTRACT

Software-Defined Networking (SDN) has the potential to automate and radically simplify management of computer networks—today a manual, error-prone task. Many networks however, especially enterprise networks, face a deployment problem: How to migrate an existing network to SDN? SDN must be introduced incrementally to build confidence and respect infrastructure budget constraints.

In this article, we present an approach to design and operate hybrid networks that combine both traditional and SDN switches. Our architecture, called Panopticon, exposes an abstraction of a logical SDN programming interface for incrementally deployable software-defined networks, where SDN benefits can extend over the entire network. Based on network simulation and emulation experiments, we find that the SDN capabilities can be realized even when the SDN deployment covers a small fraction of the entire network.

1. INTRODUCTION

Although Software-Defined Networking (SDN) promises principled approaches to longstanding network operations problems, the transition of existing networks to SDN will not be instantaneous. With the exception of a few notable real-world deployments, e.g., Google's software-defined WAN [4], SDN remains largely an experimental technology for most organizations. As such, hybrid networks—networks combining SDN and traditional network devices—are increasingly viewed as a transition path towards SDN adoption; yet research focusing on these environments has so far been modest. Hybrid networks posses practical importance [1], are likely to be a problem that will span years, and present a host of notable challenges:

- **Offering clear, immediate benefits.** The benefits of SDN should be realized as of the first deployed switch. Consider the example of Google's software-defined WAN [4], which required years to fully deploy, only to achieve benefits after a complete overhaul of their switching hardware. For enterprises, it is undesirable, and we argue, unnecessary to completely overhaul the network infrastructure before realizing benefits from SDN. An earlier return on investment makes SDN more appealing for adoption.

- **Eliminating disruption while building confidence.** Network operators must be able to incrementally deploy SDN technology in order to build confidence in its reliability and familiarity with its operation. Without these, it is risky and undesirable to replace all production control protocols with an SDN control plane as a single "flag-day" event, even if existing deployed switches already support SDN programmability. To increase its chances for successful adoption, any network control technology, including SDN, should allow for a small initial investment in a deployment that can be gradually widened to encompass more and more of the network infrastructure and traffic.

- **Respecting budget and constraints.** Network upgrade starts 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.

We argue that an appealing approach to dealing with these challenges is to abstract a hybrid network into a *logical SDN*—conceptually, a programmatic interface that exposes the network as if it were a full SDN deployment. To demonstrate this approach, we present Panopticon, an enterprise network architecture that facilitates the migration to SDN by realizing an SDN control plane for incrementally deployable software-defined networks.

2. THE PANOPTICON APPROACH

Panopticon[1] realizes a programming interface for a hybrid network by exposing the abstraction of a logical SDN. In particular, given a partial deployment of SDN switches into an existing network, Panopticon allows network operators to abstract away the traditional network devices and operate the network as an SDN comprised of just the SDN-capable switches. Using this

[1]See [5] for a more detailed and technical description.

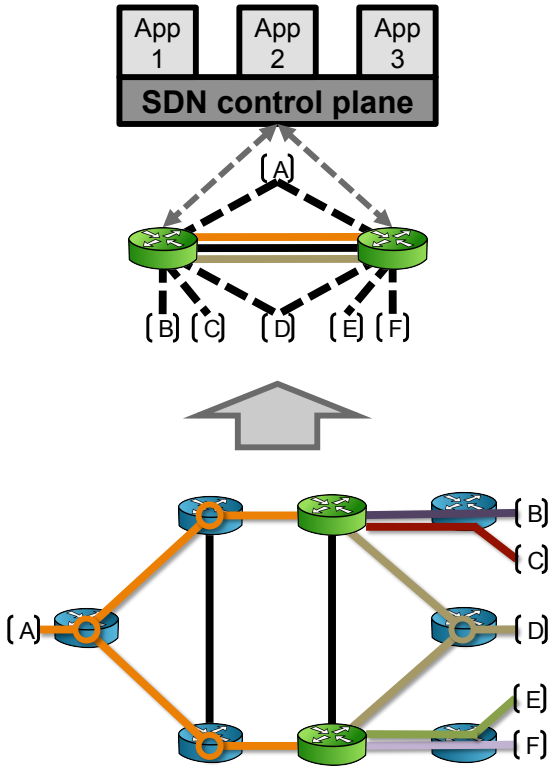

Figure 1: Overview of Panopticon. On the bottom part of the figure, an example hybrid network of 8 switches is shown: SDN switches are depicted in green, and traditional switches are depicted in blue and overlaid with the SCTs (Solitary Confinement Trees) of every SDNc port from $[A]$ through $[F]$. Each SCT is realized with a different VLAN ID, represented by using different colors. The top part of the figure illustrates the corresponding logical SDN in which all SDNc ports are virtually connected to SDN switches via "pseudo-wires".

approach, with careful planning of the partial SDN deployment, SDN capabilities can be extended to potentially every switchport in the network, not just the ports of SDN switches. Alternately, not every port need be controlled through the SDN interface, and in practice, resource constraints in the network may prevent a full SDN abstraction.

Our architecture works on the principle that every packet in the network that traverses at least one SDN switch can have the end-to-end network policy (e.g., access control) applied to it via the SDN programming interface. Moreover, traffic that traverses two or more SDN switches may be controlled at finer levels of granularity enabling further customized forwarding (e.g., load-balancing). Thus, Panopticon extends SDN capabilities to traditional switches by ensuring that all traffic to or from any operator-selected, SDN-controlled (SDNc) port is always restricted to a safe end-to-end path, that is, a path that traverses at least one SDN switch. We call this property *Waypoint Enforcement*.

Panopticon uses VLANs to restrict forwarding on traditional network devices and guarantee Waypoint Enforcement, as VLAN capabilities are ubiquitously available on existing switches. However, since the VLAN ID space is limited to 4096 values, and often fewer are supported in hardware, we devise a scalable Waypoint Enforcement mechanism called the *Solitary Confinement Tree* (SCT). An SCT corresponds to a spanning tree connecting an SDNc port to certain SDN switches. As such, each SCT provides a safe path from an SDNc port to every SDN switch it connects to. A single VLAN ID is assigned to each SCT, which ensures traffic isolation, and provides per-destination path diversity. The scalability of this approach stems from the fact that VLAN IDs can be reused for disjoint SCTs, that is, SCTs that do not traverse a common traditional network device.

To illustrate through example, consider the hybrid network of eight switches show in Figure 1 (*bottom*). In this example, the SCT of SDNc port $[A]$ is the tree that consists of the links depicted in orange; similarly, the SCTs of other ports are depicted each with a different color. Figure 1 (*top*) shows the corresponding logical SDN of the physical hybrid network enabled by SCTs (a.k.a. VLANs). In this logical SDN, every SDNc port is connected to at least one SDN switch via a "pseudo-wire" (realized by its SCT).

3. FEASIBILITY AND OVERHEADS

The logical SDN abstraction does not come for free, as the Waypoint Enforcement of traffic through SDN switches can lead to increased path lengths and link utilizations in some cases. Consequently, Panopticon presents operators with various resource-performance trade-offs, e.g., between the size and fashion of the partial SDN deployment, and the consequences for the traffic. However, Panopticon also introduces new opportunities to improve traffic control within the network, e.g., enabling multi-path forwarding for load-balancing when sufficient path diversity exists.

To understand the feasibility of our approach, we simulate different partial SDN deployment scenarios using a large campus network topology under different resource constraints and traffic conditions. (For more details on the methodology, see [5].) Concretely, the topology consists of roughly 1700 switches. These simulations let us (*i*) evaluate the feasibility space of our architecture, (*ii*) explore the extent to which SDN control extends to the entire network, and (*iii*) understand the impact of partial SDN deployment on link utilization and path stretch.

Figure 2 (*left*) illustrates that the ability to accommodate more SDNc ports with a small number of SDN switches depends largely on the number of VLAN IDs supported for use by the traditional hardware. Under favorable conditions with 1024 VLANs, feasibility for 100% SDNc ports requires as few as 33 SDN switches. VLAN ID availability is necessary to construct SCTs

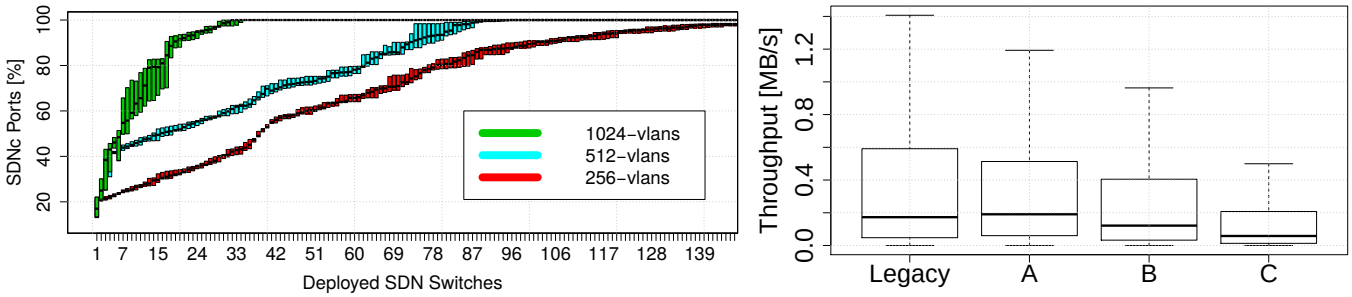

Figure 2: ***Left:*** **Percentage of SDNc ports vs. number of deployed SDN switches.** ***Right:*** **Effects on TCP Throughput.**

and we see that when traditional switches support at most 256 VLANs, over 140 SDN switches must be deployed before achieving full SDNc port feasibility.

To compliment our simulation-based approach and further investigate the consequences of Panopticon on traffic, we conduct a series of emulation-based experiments on portions of a real enterprise network topology.

Figure 2 (*right*) illustrates the impact of Waypoint Enforcement on TCP performance in three scenarios: in Scenario A, 28 switches are operated as SDN switches, and Scenarios B and C narrow down the number of SDN switches to 10 and 5, respectively. SDN switch locations are selected at random based on the location of IP routers in the original topology dataset.

We see that in Scenario A, the impact on median TCP throughput is small. This is perhaps expected, as all traffic across subnets must traverse some IP router in the legacy network, regardless. Some flows experience congestion due to Waypoint Enforcement. Other flows actually experience a performance increase due to the availability of multiple alternate paths in Panopticon, which cannot be exploited in a traditional Ethernet-based network. As the SDN deployment shrinks to more conservative sizes in Scenarios B and C, the effects of Waypoint Enforcement become more prominent, supporting our observed simulation results.

4. CONCLUDING REMARKS

We view our work as a concrete step towards systematic, incremental deployment for SDN. Accordingly, we have presented our work at the IRTF Working Group on SDN and we plan to contribute our results to the ongoing discussions at the Migration Working Group of the Open Networking Foundation [1].

Our work contributes to a field that is attracting increasing attention from other researchers. Agarwal *et al.* [2] demonstrate effective traffic engineering for traffic that crosses at least one SDN switch in a partial deployment. Panopticon is an architecture that enforces this condition for all SDNc ports. The work on software-controlled routing protocols by Vanbever and Vissicchio [6] presents mechanisms to enable an SDN controller to indirectly program L3 routers by carefully crafting routing messages. We view this work as complementary to ours in that it could be useful to increase control over traffic whose paths include IP routers. Hand and Keller [3] propose an alternate approach called ClosedFlow that aims to enable SDN control over existing proprietary hardware by mimicking the fine grain control available in OpenFlow. Vissicchio *et al.* [7] discuss certain trade-offs that arise within a diverse set of hybrid SDN models and argue that hybrid SDN architectures deserve more attention from the scientific community. We agree and are hopeful that our work will offer a helpful reference point for practical hybrid Software-Defined Networking and contribute to ongoing standardization efforts.

5. REFERENCES

[1] Open Networking Foundation Migration Working Group: Migration Use Cases and Methods. `https://www.opennetworking.org/images/stories/downloads/sdn-resources/use-cases/Migration-WG-Use-Cases.pdf`.

[2] S. Agarwal, M. Kodialam, and T. V. Lakshman. Traffic Engineering in Software Defined Networks. In *INFOCOM*, 2013.

[3] R. Hand and E. Keller. ClosedFlow: OpenFlow-like Control over Proprietary Devices. In *HotSDN*, 2014. To appear.

[4] S. Jain et al. B4: Experience with a Globally-Deployed Software Defined WAN. In *SIGCOMM*, 2013.

[5] D. Levin, M. Canini, S. Schmid, F. Schaffert, and A. Feldmann. Panopticon: Reaping the Benefits of Incremental SDN Deployment in Enterprise Networks. In *USENIX ATC*, 2014.

[6] L. Vanbever and S. Vissicchio. Enabling SDN in Old School Networks with Software-Controlled Routing Protocols. In *Open Networking Summit (ONS)*, 2014.

[7] S. Vissicchio, L. Vanbever, and O. Bonaventure. Opportunities and Research Challenges of Hybrid Software Defined Networks. *ACM Computer Communication Review*, 44(2), April 2014.