HARMLESS: Cost-Effective Transitioning to SDN

Márk Szalay
Budapest University of Technology and Economics
szalay@tmit.bme.hu

László Toka
MTA-BME Information Systems Research Group
toka@tmit.bme.hu

Gábor Rétvári
Budapest University of Technology and Economics
retvari@tmit.bme.hu

Gergely Pongrácz
Ericsson Research,
TrafficLab Hungary
gergely.pongracz@ericsson.com

Levente Csikor
School of Computing Science,
University of Glasgow
levente.csikor@glasgow.ac.uk

Dimitrios P. Pezaros
School of Computing Science,
University of Glasgow
dimitrios.pezaros@glasgow.ac.uk

ABSTRACT

Recently, Software-Defined Networking has grown out of being an "intriguing approach" and turned into a "must-have" for communication networks to overcome many long-standing problems of traditional networking. However, there are still some obstacles on the way to the widespread adoption. Current commodity-off-the-shelf (COTS) SDN offerings are still in their infancy and are notorious for lacking standards compliance, scalability, and unpredictable performance indicators compared to their legacy counterparts. On the other hand, recent software-based solutions might mitigate these shortcomings, but in terms of cost-efficiency and port density they are in a lower league.

Here, we present HARMLESS, a novel SDN switch design that combines the rapid innovation and upgrade cycles of software switches with the port density of hardware-based appliances into a fully data plane-transparent, vendor-neutral and cost-effective solution for smaller enterprises to gain a foothold in this era. The demo showcases the SDN migration of a dumb legacy Ethernet switch to a powerful, fully reconfigurable, OpenFlow-enabled network device without incurring any major performance and latency penalty, nor any substantial price tag enabling to realize many use cases that would have otherwise needed standalone hardware appliances.

CCS CONCEPTS

• **Networks → Routers**; *Bridges and switches*; Layering;

KEYWORDS

Software-Defined Networking, Switch design, Migration

ACM Reference format:
Márk Szalay, László Toka, Gábor Rétvári, Gergely Pongrácz, Levente Csikor, and Dimitrios P. Pezaros. 2017. HARMLESS: Cost-Effective Transitioning to SDN. In *Proceedings of SIGCOMM Posters and Demos '17, Los Angeles, CA, USA, August 22–24, 2017,* 3 pages.
DOI: 10.1145/3123878.3123882

SIGCOMM Posters and Demos '17, Los Angeles, CA, USA

DOI: 10.1145/3123878.3123882

1 INTRODUCTION

Software-Defined Networking (SDN) radically changes the way we build and operate data communication networks: decoupling the control and data plane functionality enables the traffic steering logic to be directly programmable, while allowing the underlying infrastructure to be abstracted for different network applications and services like automated orchestration, on-the-fly chaining of services, and multi-tenancy support in an "as-a-service" manner [7, 8, 11, 12, 15, 16]. Over the past few years, SDN has grown from an "intriguing approach" to a "must-have" for communication networks, in order to overcome many long-standing problems of traditional networking [16].

However, as the low number of actual SDN deployments [6] shows, small enterprises without substantial in-house expertise face significant business and technical barriers when migrating to SDN. Each *SDN migration strategy* [12] involves different *cost, management*, and *performance trade-offs* that need tedious *forward planning* and *extensive investigation* of vendor device options. The incremental migration approach offers less interference with daily operation, but managing heterogeneous network architectures is painstaking. Jumping right to full-blown SDN, on the other hand, by swapping all legacy networking appliances to SDN-capable devices overnight, may mitigate the aforementioned challenge, but this is hardly an option for small businesses due to the huge capital expenditure, the flag-day deployment, and the induced service downtime.

Today, there are essentially two choices for obtaining an SDN-enabled device: buying a *COTS hardware switch* or relying on *general purpose servers and running software switches* on top. Traditionally, COTS SDN devices have been praised for providing high port density at a reasonable price tag, albeit notorious for not complying to standards, not scaling, and offering unpredictable performance [13] compared to their legacy counterparts. On the contrary, software-based solutions offer better programmability and standards-compliance, but struggle to match the port density of COTS switches due to the physical limits of the blade form factor.

In order to reconcile software and hardware switching and to assist smaller enterprises in their initial phase of SDN deployment, in this demo we present HARMLESS, a *Hybrid ARchitecture to Migrate Legacy Ethernet Switches to SDN*. HARMLESS introduces an additional level of abstraction on top of the conventional control plane–data plane separation: practical packet processing hardware is further decoupled from the switch operating system, and the OpenFlow (OF) capability is provided by a dedicated software

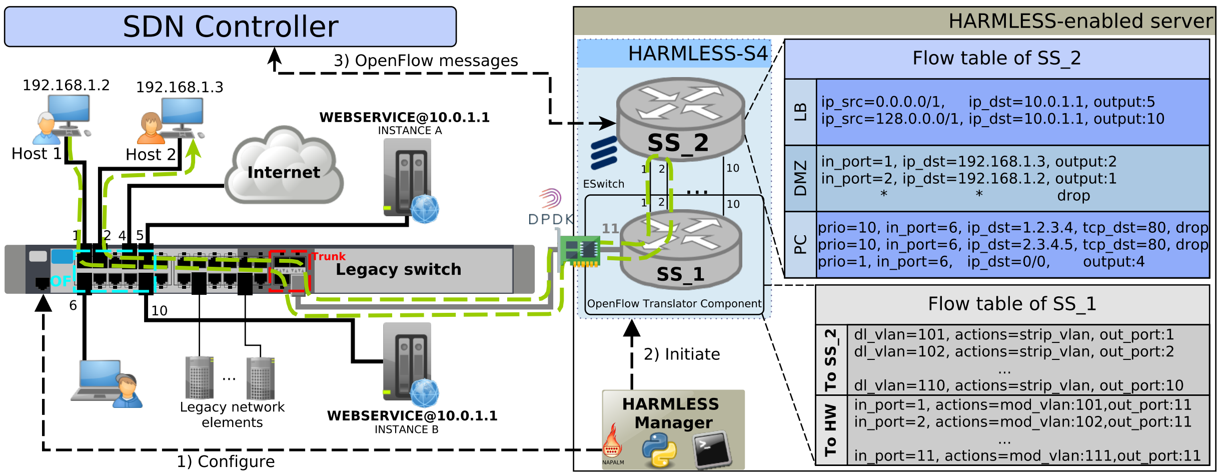

Figure 1: The HARMLESS architecture with the showcased use cases

switch. HARMLESS brings new life to plain old legacy Ethernet switches by adding SDN capability, which not only protects current investment by offering a cost-effective migration strategy but also realizes a delicate sweet spot in the trade-off between hardware simplicity and flexible functionality [1, 2, 4, 5, 10, 14].

2 ARCHITECTURE

To manage the physical ports of a legacy switch from an external software switch, we adopt the "Tagging and Hairpinning" technique, widely used to offload VM-to-VM communication from the hypervisor to special-purpose hardware at the first hop switch [5].

In our architecture (depicted in Fig. 1), the legacy switch is configured to tag each packet with a unique VLAN id that identifies the access port it was received from. Then, the tagged packets are forwarded to the software switch along the trunk-port–soft-switch interconnect to enforce the network-wide policies according to the OF program set up for the switch by the controller[1]. Since each packet goes through the OF pipeline, there is no limitation on the desired packet forwarding policy. Finally, packets are sent back to the legacy switch "hairpinned", i.e., tagged with the unique VLAN id of the proper outgoing port. To avoid having to tailor controller programs to the way HARMLESS maps output ports to VLAN ids and vice versa, we introduce an additional OpenFlow Translator Component as an adaptation layer, implemented by another software switch instance (SS_1). The latter is connected to the main OF switch (SS_2) by as many patch ports as the number of managed access ports of the legacy device in order to dispatch packets to and from the patch ports based on the used VLAN ids (see the group node *HARMLESS-S4* and *Flow table of SS_1* in Fig. 1).

As an example, consider the case of *Host 1* and *Host 2* (connected to access ports 1 and 2 identified by VLAN id 101, and 102) permitted to exchange traffic only with each other. When Host 1 sends a packet to Host 2, this is tagged with *VLAN id 101* and forwarded to SS_1 via the trunk port. According to its flow table, SS_1 outputs the packet to patch port 1, through which the main OF switch (SS_2), managed directly by the SDN controller, receives it and processes it according to the OF pipeline. Based on the policy (marked by row *DMZ*), SS_2 passes the packet back to SS_1 via patch port 2. SS_1 subsequently tags the packet with *VLAN id 102* and immediately passes it towards the legacy switch which in turn removes the VLAN tag and sends the packet to Host 2 (see green dashed arrow). Relying on Python and BASH, we developed the HARMLESS Manager [2] that automatically manages and queries the legacy Ethernet switch via SNMP through NAPALM, a library supporting numerous networking operating systems (e.g., Cisco IOS, Arista EOS). According to the desired OpenFlow-enabled port-setting, the manager configures the legacy switch, then instantiates HARMLESS-S4. Finally, it installs the corresponding flow rules into SS_1 and connects SS_2 to the SDN controller.

During the demo, we will showcase how HARMLESS renders a dumb legacy Ethernet switch to a powerful, fully reconfigurable network device without incurring any major performance [3] or latency penalty, or any substantial price tag, henceforth allowing to realize many use cases *in-network* (see Fig.1) that would otherwise need dedicated standalone hardware appliances. In the *a)* Load Balancer (LB) scenario, HARMLESS will allow to equally distribute ingress web traffic between multiple backends based on matching of the source IP address; *b)* the DMZ use case will show how to implement and fine-tune VM-level access policies in a multi-tenant cloud using OF; and *c)* the Parental Control (PC) scenario will show how to selectively deny access to specific users to certain web pages on-the-fly. Video showcase is available at https://goo.gl/ssw5Bb.

Acknowledgments The work has been supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) projects EP/L026015/1, EP/N033957/1, EP/P004024/1, by the European Cooperation in Science and Technology (COST) Action CA 15127: RECODIS – Resilient communication services protecting end-user applications from disaster-based failures, and by Ericsson. Laszló Toka was partially funded by Bolyai Scholarship of the Hungarian Academy of Sciences.

[1] The OF components can be realized by any software switch, we used ESwitch [9] with DPDK-enabled NICs.

[2] http://github.com/muuurk/harmless

REFERENCES

[1] K. Argyraki, S. Baset, B.-G. Chun, K. Fall, G. Iannaccone, A. Knies, E. Kohler, M. Manesh, S. Nedevschi, and S. Ratnasamy. Can software routers scale? In *PRESTO*, pages 21–26, 2008.

[2] M. Casado, T. Koponen, D. Moon, and S. Shenker. Rethinking packet forwarding hardware. In *HotNets*, pages 1–6, NY, USA, 2008. ACM.

[3] L. Csikor, M. Szalay, B. Sonkoly, and L. Toka. NFPA: Network function performance analyzer. In *IEEE Conference on Network Function Virtualization and Software Defined Networks Demo Track (NFV-SDN)*, pages 17–19, San Francisco, CA, USA, 2015, http://nfpa.tmit.bme.hu.

[4] G. Ferro. Soft switching fails at scale. EtherealMind, 2011.

[5] J. Gross, A. Lambeth, B. Pfaff, and M. Casado. The rise of soft switching, Part I, II, III. Network Heresy, 2011.

[6] S. Jain et al. B4: Experience with a globally-deployed software defined wan. In *SIGCOMM*, pages 3–14, NY, USA, 2013. ACM.

[7] 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*, pages 333–345, 2014.

[8] M. McNickle. With hybrid SDN deployment, no need for network forklift. TechTarget SearchSDN, 2013.

[9] L. Molnár, G. Pongrácz, G. Enyedi, Z. L. Kis, L. Csikor, F. Juhász, A. Kőrösi, and G. Rétvári. Dataplane specialization for high-performance openflow software switching. In *SIGCOMM*, pages 539–552, Florianopólis, 2016. ACM.

[10] D. Moon, J. Naous, J. Liu, K. Zarifis, M. Casado, T. Koponen, S. Shenker, and L. Breslau. Bridging the software/hardware forwarding divide. UC Berkeley.

[11] B. S. Networks. Migrating from a Traditional Network to a Software-Defined Network. https://www.sdxcentral.com/articles/featured/big-switch-state-of-sdn-adoption/2016/09/, Sept 2016.

[12] Open Networking Foundation. SDN Migration Considerations and Use Cases. ONF Solution Brief, 2014.

[13] I. Pepelnjak. Q&A: Vendor Openflow Limitations. http://blog.ipspace.net/2016/12/q-vendor-openflow-limitations.html, Dec 2016.

[14] G. Pongrácz, L. Molnár, Z. L. Kis, and Z. Turányi. Cheap silicon: A myth or reality? picking the right data plane hardware for software defined networking. In *HotSDN*, pages 103–108, 2013.

[15] S. Vissicchio, O. Tilmans, L. Vanbever, and J. Rexford. Central control over distributed routing. In *SIGCOMM*, pages 43–56, 2015.

[16] Zeus Kerravala. The top five reasons to deploy software-defined networks and network functions virtualization. Report: https://goo.gl/KeTEcd, May 2014.