PINT-BoX: Path-aware networking IN a Tofino BoX

Everson S. Borges[*] [†], Willen B. Coelho[*] [†], Fabricio R. Cesen[‡], Francisco G. Vogt[‡], Christian Rothenberg[‡], Rodolfo S. Villaça[*], Cristina K. Dominicini[†], Rafael S. Guimarães[†], and Magnos Martinello[*]
[*]Department of Informatics, Federal University of Espírito Santo, Brazil.
[†]Department of Informatics, Federal Institute of Espírito Santo, Brazil
[‡]School of Electrical and Computing Engineering, Universidade Estadual de Campinas, Brazil

***Abstract*—This demonstration showcases PINT-BoX, a Path-Aware Networking using the PolKA source routing approach, in combination with the P7 emulation experimentation tool built on a Tofino box. It is designed for experimentation on a Tofino-based system, demonstrating the creation of various network topologies emulated within a P4-programmable switch. The demo highlights the flexibility and capabilities of the platform in exploring diverse network paths and configurations. The proposed design gives endpoints full control over multiple paths within a given topology, each with distinct network metrics, allowing them to change the path without the need to reconfigure the network core.**

***Index Terms*—P4, P7, PolKA, Tofino, Path-Aware Networks, Path Verification**

I. Introduction

As networking environments grow increasingly complex, driven by the demands of emerging applications such as 5G and augmented reality, the need for experimental validation before deployment becomes ever more critical [1]. Network testbeds play a pivotal role in fostering innovation and exploring new paradigms, yet creating testbeds that are accessible, cost-effective, and capable of delivering line-rate performance with precise evaluations remains a significant challenge [2]. Researchers often face resource constraints that limit the scale, realism, and quality of their experiments, forcing them to rely on smaller setups or emulation and simulation platforms.

This limitation becomes particularly evident with innovative internetworking paradigms such as path-aware architecture [3]. These architectures can significantly enhance application responsiveness and network performance by providing endpoints with path characteristics, enabling applications to make optimization decisions when selecting paths. To realize this, three key functionalities are required: i) delivering path information, ii) enabling path selection, and iii) verifying paths. Unfortunately, current network testbeds do not fully support these requirements, underscoring the need for more advanced tools capable of effectively handling experimentation with path-aware networks. This highlights the broader challenge of ensuring network testbeds meet the demands of emerging technologies and innovations.

To address this gap, we propose ***PINT-BoX***, which combines the programmability of P4 with path-aware functionalities, that supports experiments in which endpoints retrieve path information, select different paths, and verify those selections. Using a hardware-based emulation tool P7 [4] for high-fidelity experimentation with path-aware networks on a single Tofino switch. As illustrated in Figure 1, powered by P7, *PINT-BoX* provides a user-friendly environment in which researchers can write scripts, similar to Mininet, to create packet processing pipelines that emulate specific characteristics of the network link and instantiate a network topology capable of running line-rate traffic on a single physical P4 switch.

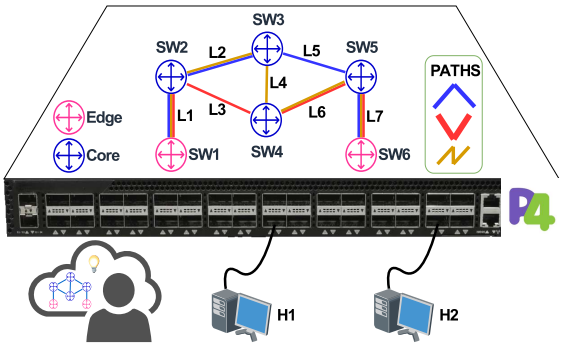

Fig. 1. *PINT-BoX* Concept, Components and Topology Representation

II. *PINT-BoX* Architecture and Demo

PINT-BoX extends P7 emulation tool[1] by integrating support to path-aware networks based on the PolKA source routing approach [5]. *PINT-BoX* introduces a novel approach by employing PolKA-based packet forwarding, which leverages remainder-of-division logic using CRC8 hardware. This innovative source routing method results in stateless core switches while providing path-aware functionality at the endpoints.

In a nutshell, P7 transforms a P4 hardware device into an emulated path-aware network capable of realistically representing various scenarios. Additionally, we improved the path verification process by utilizing the TTL (Time to Live) field to track and validate the packet's route, enabling more detailed route analysis and accurate path validation. This approach is a simple example of customized user P4 code integration.

Experimental setup. Figure 1 shows a demo topology with three available paths: $Path1$**: SW1, 2, 3, 5, 6**, $Path2$**: SW1, 2, 4, 5, 6**, and $Path3$**: SW1, 2, 3, 4, 5, 6**. These paths are exposed to physical endpoints $H1$ and $H2$. Depending on the interface specification, these endpoints can be directly connected at 10G, 25G, 40G, or 100G speeds. Each link is labeled with specific characteristics, such as bandwidth, latency, and packet loss.

Introduced in [6], the TTL can serve as a metric to validate a specific path within an emulated environment. Each switch in the topology has a specific TTL weight in Table I. The

[1]https://github.com/intrig-unicamp/p7/wiki/5)-PINT-BOX

TABLE I
TABLE WITH THE TTL WEIGHTS OF EACH CORRESPONDING SWITCH

TTL Weight	SW1	SW2	SW3	SW4	SW5	SW6
Request	4	5	6	7	8	9
Reply	10	11	12	13	14	15

TABLE II
CUSTOMIZED NETWORK LINK METRICS FOR THE DEMO IN EACH PATH.

Path	Link	Bandwidth	Loss	Latency	Cumulative TTL	
					Request	Reply
Path 1	L5	10G	3%	5 ms	32	62
Path 2	L3	10G	1%	15 ms	33	63
Path 3	L4	10G	2%	10 ms	39	75

default TTL=64 is incremented by each switch according to the weight. Table II presents the cumulative TTL weights at each path, providing a clear and comprehensive assessment of the path verification process for this topology.

Table I displays the TTL weights for six switches (SW1 to SW6), with the "Request" row for outgoing packets and the "Reply" row for returning packets. These weights are essential for path tracking and accurate packet delivery. Table II shows the cumulative TTL as the sum of individual TTL weights for each switch along the path.

Loss and Latency Experiment. Each link in the topology is configured to operate at 10Gbps, in addition, L3, L4, and L5 are characterized by customized packet loss and latency metrics. This results in three paths with distinct link characteristics, as shown in Table I. Figure 2 presents the average results of Round-Trip Time (RTT) and packet loss measurements across the three network paths. $Path1$ experiences a latency of approximately 10 ms and 6% of loss due to $L5$. $Path2$ exhibits the lowest overall packet loss but has the highest latency, primarily influenced by link $L3$. $Path3$ records a latency of 20 ms and a packet loss rate of 4%, mainly due to $L4$.

During the Demo. Three distinct network paths are used, allowing attendees to observe how *PINT-BoX* sets up an experiment by generating all necessary artifacts. The demo showcases its ability to create a testbed that replicates link characteristics. Attendees can select a path based on link metrics and dynamically switch between paths by updating the source. Real-time traffic visualization confirms the link metrics and emulation performance.

Figure 3 presents a traffic steering experiment where the endpoint selects the path based on throughput. Initially, traffic is routed through $Path1$, demonstrating the highest throughput. Then we move on to $Path2$, which has slightly lower throughput, and finally to $Path3$. This process highlights the system's capability to identify and adapt to different paths, source selected over time. Path changes with PolKA solution are instantaneous, enabling seamless transitions by requiring only an update for the correct route label at the origin of the traffic. There is no need for internal state adjustments or reconfiguration of the network core. Additionally, Figure 3 highlights the behavior of the TTL field, which changes as the path varies.

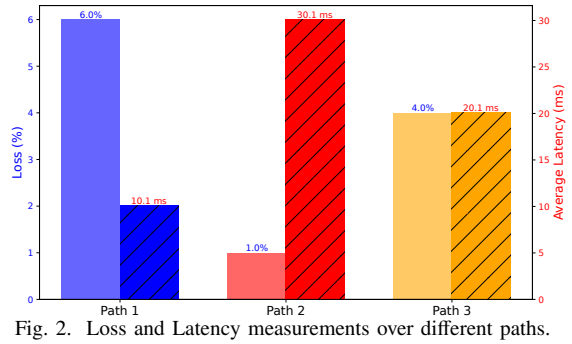

Fig. 2. Loss and Latency measurements over different paths.

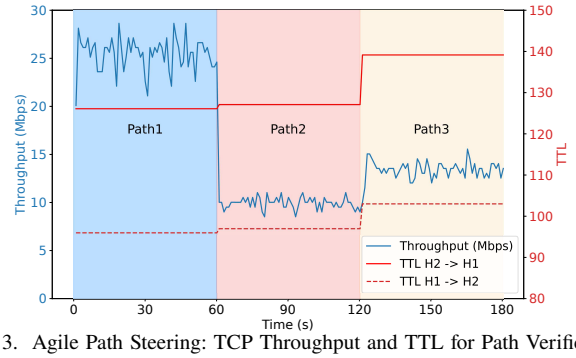

Fig. 3. Agile Path Steering: TCP Throughput and TTL for Path Verification.

III. CONCLUSION AND FUTURE WORK

This demo video [2] highlights how *PINT-BoX* enhances the ecosystem of affordable hardware-based experimental platforms as a powerful emulation tool for high-fidelity experimentation of Path-Aware Networks. As a programmable high-fidelity testbed, *PINT-BoX* enables reproducible research experiments by sharing the configuration files, which can be compiled, deployed, and validated. The *PINT-BoX* roadmap includes plans to focus on path validation, ensuring that packets follow the correct sequence of switches and adhere to the designated route. This will be important within disaggregated network initiatives such as OpenRAN.

ACKNOWLEDGEMENTS

Porvir-5G Research Project (Grant 20/05182-3), and Fapes (941/2022, 732/2024). Also, this work was supported by Ericsson Telecomunicações Ltda and by the Sao Paulo Research Foundation (FAPESP), grant `2021/00199-8`, CPE SMARTNESS. This study was partially funded by *CAPES*, Brazil - Finance Code `001` and CNPq fellow (Grant 312058/2023-3).

REFERENCES

[1] J. Smith and A. Lee, "5G and Augmented Reality: Emerging Technologies and Challenges," *Journal of Emerging Networks*, pp. 35–45, 2020.

[2] I. Baldin *et al.*, "Fabric: A national-scale programmable experimental network infrastructure," *IEEE Internet Computing*, 2019.

[3] B. Trammell, "Current Open Questions in Path-Aware Networking," RFC 9217, Mar. 2022.

[4] F. Rodriguez *et al.*, "P4 programmable patch panel (P7): An instant 100G emulated network on your Tofino-based pizza box," in *SIGCOMM'22 Poster and Demo Sessions*, 2022, pp. 4–6.

[5] C. Dominicini *et al.*, "PolKA: Polynomial Key-based Architecture for Source Routing in Network Fabrics," in *NetSoft*, 2020, pp. 326–334.

[6] F. E. R. Cesen *et al.*, "Towards multiple pipelines network emulation with p7," in *NetSoft*, 2023, pp. 290–292.

[2]https://drive.google.com/file/d/1G97-_B2_gxrlP17GXGTCHJuua6V9axie