Path-Aware Network with Residue Number System : Intrinsic Path Verification and High-Performance Routing

Everson Scherrer Borges[1 2], **Magnos Martinello**[2], **Cristina Klippel Dominicini**[1], **Moises R. N. Ribeiro**[2]

[1]Instituto Federal do Espirito Santo (IFES)

[2]Universidade Federal do Espirito Santo (UFES)

{everson@ifes.edu.br}

Abstract. *The Internet's end-to-end design keeps routers simple by delegating complexity to end hosts, but provides little or no path verification. As a result, many available paths remain unused, their quality unknown, and risks such as traffic deviation or forwarding errors may arise. Effective path awareness requires three essential properties: (i) Verifiability ensuring packets traverse the intended path; (ii) Controllability enabling explicit path selection; and (iii) Visibility providing path information to support routing decisions. Existing approaches to path verification impose significant computational overhead and cannot be efficiently deployed on programmable switches, relying on mechanisms such as origin tracing, chained MACs, or nested cryptographic structures that are not natively integrated into the routing process. This work introduces a path-aware network architecture that combines source routing with the Residue Number System (RNS). A routeID encodes the entire path and is decoded at each hop via modular operations. Beyond routing, the routeID enables intrinsic path verification: it serves as a key for Proof-of-Transit (PoT) lookups — extending the IETF RFC draft via Mersenne-based Shamir Secret Sharing — and, in a second tableless design, drives a chain of hash operations forming a path signature, a property unattainable in existing routing systems.*

1. Introduction

A historical analysis of path awareness in network design highlights two main paradigms: intelligent routers paired with simple hosts, and intelligent hosts relying on simple routers. The first approach, in which routers handle path selection and network intelligence, was ultimately abandoned due to the minimalist principles of Internet design. Instead, the dominant model follows the end-to-end argument, placing complexity at the network edges endhosts manage reliability, packet loss, and congestion control, while routers maintain simplicity by only forwarding packets. This model laid the foundation for the modern Internet architecture.

Although effective, the Internet lacks native path awareness. The design of the IP protocol was not intended to provide path knowledge, as it was conceived primarily as a system to ensure connectivity between devices, without considering the specifics of the underlying route [Postel 1981]. This simplicity allowed the routers simply forward packets based on destination addresses without keeping track of the specific path. However, this lack of path awareness has become a limitation in modern networks, where the

need for more control over routing, path selection, and performance metrics has become increasingly important [Jia et al. 2020].

In this model, path verification is largely absent, preventing applications from confirming or influencing the route traversed. As a result, many paths remain unused, their quality unknown, and vulnerabilities such as traffic deviation or forwarding errors may occur. To address these challenges, Path Aware Network (PAN) [Trammell 2022] architectures have been proposed to increase transparency and give to the hosts (or edges) greater control over network paths [Godfrey et al. 2009, Barrera et al. 2017, Anderson et al. 2014].

For path awareness to be effective, it must provide three fundamental properties: (i) verifiability, ensuring packets follow the intended path; (ii) controllability, enabling explicit path selection; and (iii) visibility, offering path information to support informed routing decisions. This work extended summary addresses the following **research question**: Can path verification be natively integrated into the routing process to enable high performance deployment in programmable data planes?

2. Problem Statement

The problem arises when network traffic is intentionally deviated from its intended path, violating security policies and causing forwarding inconsistencies (i.e., path deviation attacks) which are assumed to be our adversary model, as described in [Bu et al. 2020].

Skipping: Refers to a malicious routing behavior, wherein a router redirects a packet, intentionally bypassing one or more intermediary routers that are supposed to be part of the designated path. This behavior is depicted in Figure 1, where, for instance, the packet avoids router 3 and proceeds along the alternative route, passing through the path $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$.

Addition: The packet temporarily diverges from the originally intended path and subsequently rejoins it. During this process, the packet may traverse one or more routers that were not initially part of the expected route. A visual representation, in Figure 2, with the altered path: $1 \rightarrow 2 \rightarrow 6 \rightarrow 3 \rightarrow 4 \rightarrow 5$.

Path detour: Malicious router R_1 causes a packet to deviate from the intended path, but later it returns to the correct path. Partial detour is illustrated in Figure 3, where the packet deviates from some but not all routers (path $1 \rightarrow 2 \rightarrow 6 \rightarrow 7 \rightarrow 4 \rightarrow 5$). Complete detour is shown in Figure 4.

Out of Order: Malicious router (e.g., R_2) diverts a packet from its intended path before it eventually rejoins the correct sequence. As illustrated in Figure 5, the packet may skip or reorder some routers, such as following the path $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5$. Out of order is shown in Figure 5.

Path verification can be formally defined as the process of ensuring that a packet has traversed a predefined sequence of forwarding elements, in compliance with an intended routing policy[Bu et al. 2020]. In this context, the goal is to guarantee that every packet's actual forwarding path matches its prescribed path, thus preventing unauthorized deviations.

Path deviation attacks pose a serious threat to critical infrastructures such as

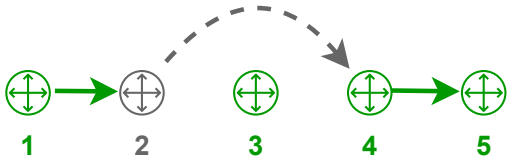

Figura 1. Skipping

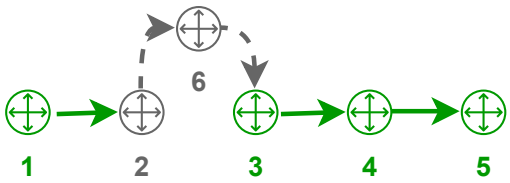

Figura 2. Addition

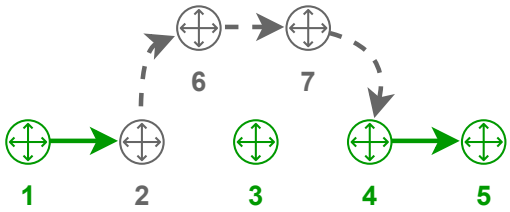

Figura 3. Partial Detour

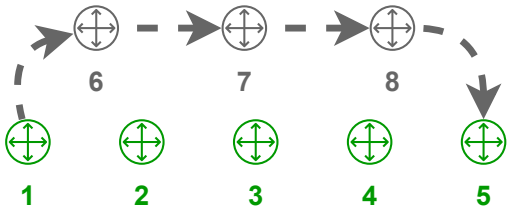

Figura 4. Complete Detour

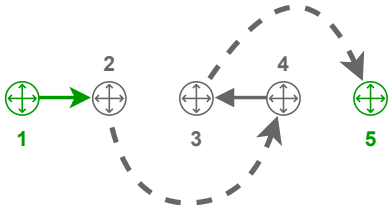

Figura 5. Out Of Order

financial transaction backbones or **sovereign paths** [Krähenbühl et al. 2023]. When compliance requires messages to follow a specific route, it is essential that edge nodes or end hosts can verify whether on-path routers actually respect these directives [Legner et al. 2020].

In such environments, *strict compliance with forwarding policies* is essential to ensure data integrity, performance isolation, and regulatory adherence. An adversary controlling a single compromised router could perform behaviors like skipping, inserting, detouring, or reordering packets, undermining trust in the network's service provider and making traditional monitoring insufficient.

State of the art: Most existing approaches specifically related to path verification, impose significant computational overhead at the source or core nodes and cannot be efficiently deployed on programmable switches. These solutions typically rely on mechanisms such as origin tracking [Naous et al. 2011], chained MACs [Zhang et al. 2014], or nested cryptographic structures [Wu et al. 2018, Legner et al. 2020] to achieve path verification. However, they are not designed to integrate path verification natively within the routing process, limiting their applicability in programmable data planes.

3. Hypotheses

This work extended summary argues that a **RNS-based source routing** [Dominicini et al. 2020] can provide intrinsic path verification and efficient routing for a path-aware network approach, overcoming the limitations of existing approaches that impose high overhead and lack integration with programmable data planes.

The novel path-aware network architecture combines source routing [Sunshine 1977] with the Residue Number System (RNS) [Szabo and Tanaka 1967] to address the fundamental security-efficiency dilemma in the data plane. The core principle is leveraging the Residue Number System (RNS) mathematical structure to integrate path verification intrinsically with minimal communication and computational overhead, achieving a viable balance suitable for deployment in resource-constrained programmable data planes.

On one hand, it employs **short per-hop verifiers**, *implemented by high-speed modulo operations* derived from irreducible polynomials in the RNS. On the other hand, it constructs **a path signature** that can be verified directly within the routing system using Proof-of-Transit (PoT) operations at each hop.

In our design, a *routeID* encodes the entire packet path and is decoded at each hop via modulo operations, guiding the packet explicitly through a sequence of nodes with their respective ports rather than simply directing it to a destination address. Beyond routing, our design enables path verifiability into the routing process. Initially, the *routeID* is used as a key to perform lookups in a Proof-of-Transit table. This extends the IETF Request For Comments (RFC) draft by integrating Mersenne numbers, enabling the practical implementation of Shamir Secret Sharing within programmable switches.

Then, a second tabless design is introduced by using the unique correspondence between a *routeID* and its constituent *nodeID*s to compute a chain of hash operations that form a *path signature*, supporting a intrinsic path verification, a property that is not achievable in existing routing systems.

4. Contributions

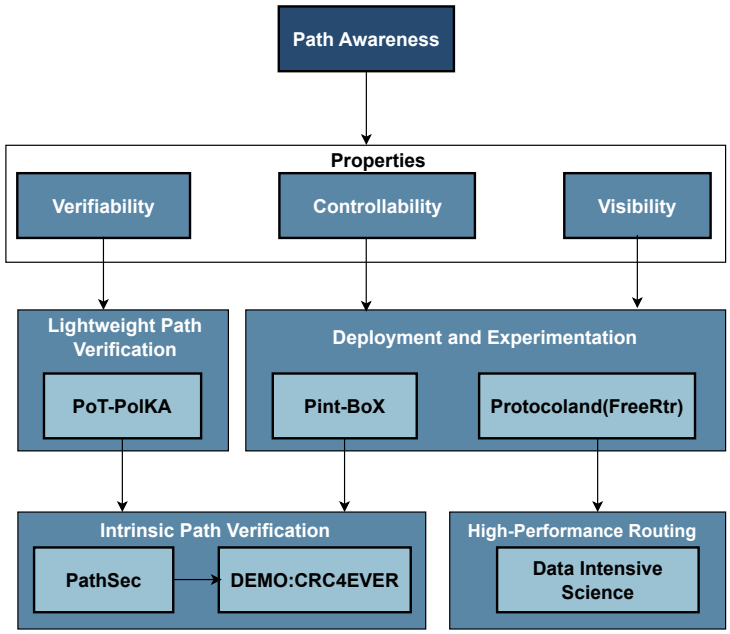

Figura 6. Overview of this work extended summary highlighting the main works.

The Figure 6 illustrates the key contributions of this work, structured around

the properties of path-aware networking. On the left side of the figure, the design of mechanisms such as PoT-PolKA [Borges et al. 2023], [Borges et al. 2024b] and Path-Sec [Martinello et al. 2024], related to the path verification. The middle and right sections highlight the contributions related to controllability and visibility, including hardware emulation tools **Pint-Box** [Borges et al. 2024a] and routing platform **FreeRtr**, which enable the practical deployment [Borges et al. 2022] with Prove of Concept (PoC) realization. At the base, the figure outlines the pillars of this work: Intrinsic Path Verification and High-Performance Routing, through Deployment and Experimentation with an Use Case[Ventorim et al. 2025] in Data-Intensive Science (DIS) and Demonstrations Pint-Box and CRC4EVER [Borges et al. 2025].

The extended summary work makes three fundamental contributions:

- **Contribution 1:** The author introduces a novel PoT-PolKA design [Borges et al. 2024b], crafted for programmable networks. This innovative design leverages PolKA source routing [Dominicini et al. 2020] to ensure strict path selection while incorporating a modified implementation of the PoT from IETF RFC draft for path verification integrated with P4-programmable switches. This work extends the IETF RFC draft by integrating Mersenne numbers, enabling the practical implementation of Shamir Secret Sharing within programmable switches.
- **Contribution 2:** The Path-Aware Secure Routing (PATHSEC) approach incorporates a native path verification mechanism using the RNS and the CRT, generating a unique multi-signature based on HMAC. This ensures accurate and reliable path verification within networks. Still in this contribution, the PINT-BoX platform is introduced as an innovative emulation tool that integrates the PolKA source routing approach. By employing remainder-of-division logic with CRC8 hardware, PINT-BoX enables stateless core switches while maintaining path awareness at endpoints, allowing flexible and dynamic network architectures.
  Complementing the path verification, the CRC4EVER prototype demonstrates an innovative way to design networking functions by using the mathematical properties of the RNS implemented by existing CRC mechanisms.
- **Contribution 3:** This contribution advances the integration of the PolKA protocol into a fully programmable routing environment, extending its control and data plane functionalities beyond the initial Proof of Concept (PoC) stage. This extension focuses on demonstrating the deployment feasibility, controllability, and visibility of the path-aware architecture, serving as a necessary complementary focus to path verification mechanisms. This transformation, critical for real-world adoption, leverages the robust and scalable FreeRtr platform within the RARE project, moving past the initial PoC which relied on manual scripts and lacked a routing operating system. By embedding PolKA into RARE's carrier-class routing ecosystem, the framework achieved coexistence with traditional protocols like Segment Routing and utilized built-in features such as encapsulation, Access Control Lists (ACLs), and importantly, Policy-Based Routing (PBR) for fine-grained traffic engineering. This robust environment specifically targets the highly demanding Data-Intensive Science (DIS) use case, validating the architectural performance through extensive evaluations across testbeds that demonstrated scalability, support for agile path reconfiguration, and capabilities for aggregate throughput beyond 200 Gbps over long-distance WANs.

5. Results

5.1. Intrinsic Path Verification by RNS semantics

The CRC4EVER architecture [Borges et al. 2025] is specifically designed for path verification and routing using a unique encoding scheme. It operates within programmable switches, leveraging CRC-based operations at its core. Figure 7 presents the conceptual design, illustrating the step-by-step packet flow through a sequence of network nodes in a path-aware network.

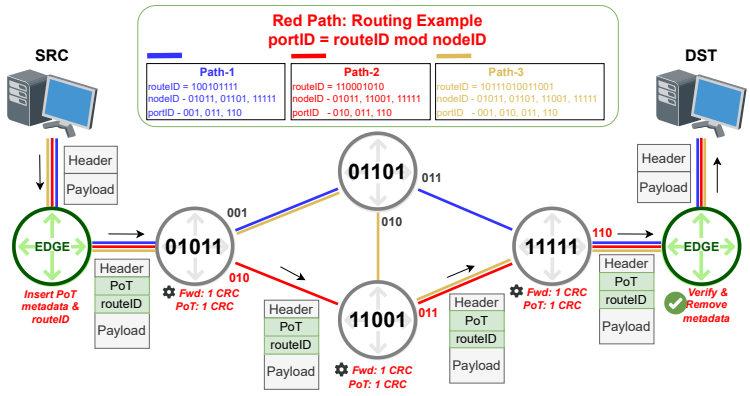

Figura 7. Conceptual Design of CRC4EVER

At the source node, a Type of Service (ToS) field is used to map the flow to its corresponding *routeID* and *PoT* metadata inserted into the packet header at the edge. For example, the blue path at the first node, the *routeID=100101111* mod *nodeID=01011* gives *portID=001*, following the computation steps below:

1. $G = \texttt{nodeID} = 01011$, so $r = \deg(G) = 3$
2. $D = \texttt{routeID} \div 2^r = 100101111 \gg 3 = 100101$ ***(SHIFT RIGHT)***
3. $\texttt{dif} = \texttt{routeID} - D \cdot 2^r = 100101111 \oplus (100101 \ll 3)$
   $= 100101111 \oplus 100101000 = 111$ ***(SHIFT LEFT, XOR)***
4. $R = \langle D \cdot 2^r \rangle_G = \langle 100101000 \rangle_{01011} = 110$ ***(CRC)***
5. $\texttt{portID} = \texttt{dif} \oplus R = 111 \oplus 110 = \boxed{001}$ ***(XOR)***

For the PoT, a unique mapping between the routeID and the sequence of node identifiers (nodeIDs) generates a path signature, as shown in Equation 1. This mechanism enables *intrinsic path verification* by supporting end-to-end PoT through chained CRC computations. At the egress edge, the path verification process validates the PoT metadata.

$$PoT_i = CRC(nodeid(i) \;||\; portid(i) \;||\; PoT_{i-1}) \tag{1}$$

Figure 8 illustrates a PoC deployment of CRC4EVER on a single core switch, demonstrating how multiple logical switches can coexist through pipeline isolation. The design splits processing into two separate pipelines: one for packet forwarding and another for PoT computation. The packet header is inserted at the ingress edge (step 1). Packets enter through the ingress pipeline, where the *routeID* and PoT header are processed. A CRC8 operation is used to decode the *routeID* by computing its modulo with the *nodeID* (step 2), determining the *portID* to the next logical switch. The packet is then recirculated and re-enters through the ingress pipeline, (step 3). In the verification pipeline, a

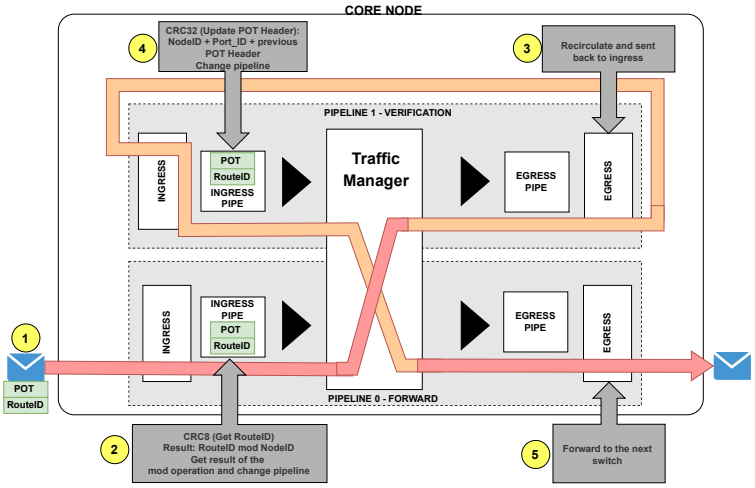

Figura 8. CRC4EVER deployment multiple logical switches in a single Tofino box

CRC32 operation updates the PoT metadata header using the Equation 1 *nodeID*, *portID*, and the previous PoT value (step 4). Finally, the packet is forwarded to the next logical switch (step 5). This architecture enables accurate forwarding and path verification entirely within a single Tofino 1 device, using the modularity of CRC operations.

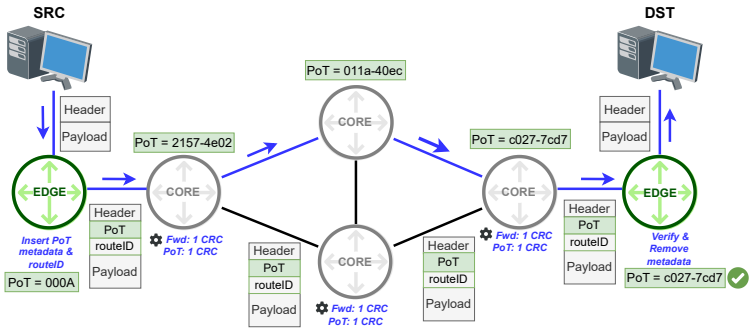

Figura 9. Experimental validation of the Path 1 forwarding process, where each hop performs CRC-based forwarding and PoT verification without modifying the packet header.

Path Verification implemented with CRC operations: The experiment illustrated in Figure 9 demonstrates the forwarding process using **Path1** within the proposed programmable data plane architecture. In this configuration, packets are generated at the source (**SRC**) and forwarded through the edge and core switches. The initial value is set to `0xA` (decimal 10). At the first core router, using Equation1, the value becomes `2157-4E02`; at the second core router, it updates to `011A-40EC`; and at the third core node, it reaches `C027-7CD7`, which represents the value to be verified according to the assigned **routeID** and PoT fields.

The PoT field is updated at every hop to indicate the current position within the path, and the metadata is finally removed at the destination (**DST**), confirming end-to-end delivery without any header modification. A video demonstration of this experiment is available online[1].

[1] `https://www.youtube.com/watch?v=RLAjzA5lC08`

5.2. Deployment in Experimental Testbeds

The proposed mechanisms were validated in real experimental infrastructures, including the Caltech P4 testbed, using programmable Tofino switches and data-intensive traffic. The topology provides multiple disjoint paths, enabling controlled evaluation of path selection, verification, and performance under realistic conditions.

5.3. Achieving Aggregate Throughput Beyond 200 Gbps

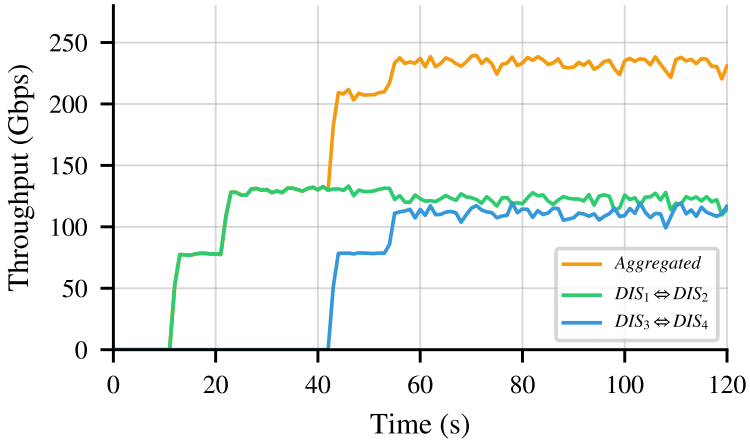

Figura 10. Throughput over time for flows between $DIS_1 \leftrightarrow DIS_2$ and $DIS_3 \leftrightarrow DIS_4$ in the PolKA testbed, demonstrating sustained high-rate data transfers

The PolKA@Caltech P4 Lab Testbed topology, illustrated in Figure 10, was designed to sustain line-rate Big Data transfers at 100 Gbps by aggregating multiple TCP flows into pre-configured PolKA tunnels. At the edge nodes, traffic steering enables flows from different sources to be directed into tunnels identified by route-ids, each representing a distinct path in the underlay. This approach allows the system to scale throughput beyond 100 Gbps through parallel aggregation, while also enabling experiments with dynamic traffic control and modulo based on reuse of CRC for packet forwarding on Tofino switches.

Figure 10 shows the throughput achieved by concurrent data-intensive flows steered over PolKA tunnels. The system sustains stable high-rate transfers and achieves aggregate throughput beyond 200 Gbps. The observed limitations are attributed to host I/O constraints rather than network forwarding, demonstrating that the proposed architecture scales to high-performance data-intensive science environments.

6. Conclusions

The Internet's end-to-end design prioritized scalability and simplicity, but left path transparency and verifiability as unresolved gaps. As modern networks demand finer traffic control and stricter compliance guarantees, the absence of native path verification mechanisms becomes both a security and a performance liability. Existing approaches that rely on cryptographic proofs or external monitoring impose overhead incompatible with high-speed programmable switches, making their deployment impractical at scale. This work addresses that gap by proposing an RNS-based source routing architecture that

integrates path verification directly into the forwarding process. Through PoT-PolKA, path compliance is enforced hop-by-hop using Mersenne-based Shamir Secret Sharing within P4-programmable switches extending the IETF RFC draft without cryptographic overhead. Through PathSec and CRC4EVER, intrinsic path verification is achieved via chained CRC operations that form a path signature natively tied to the routeID, a property unattainable in existing routing systems. Deployment on the Caltech P4 testbed validated the architecture under data-intensive conditions, sustaining aggregate throughput beyond 200 Gbps with host I/O as the only observed bottleneck. Together, these contributions redefine path awareness as a verifiable property of the data plane not merely a control-plane abstraction. The result is a foundation for secure, transparent, and high-performance programmable networks aligned with the goals of Path-Aware Networking.

Referências

Anderson, T., Birman, K., Broberg, R., Caesar, M., Comer, D., Cotton, C., Freedman, M. J., Haeberlen, A., Ives, Z. G., Krishnamurthy, A., Lehr, W., Loo, B. T., Mazières, D., Nicolosi, A., Smith, J. M., Stoica, I., van Renesse, R., Walfish, M., Weatherspoon, H., and Yoo, C. S. (2014). A brief overview of the nebula future internet architecture. *SIGCOMM Comput. Commun. Rev.*, 44(3):81–86.

Barrera, D., Chuat, L., Perrig, A., Reischuk, R. M., and Szalachowski, P. (2017). The scion internet architecture. *Commun. ACM*, 60(6):56–65.

Borges, E., Rodriguez, F., Guimarães, R. S., Martinello, M., Dominicini, C. K., Ribeiro, M. R. N., Marin, E., and Rothenberg, C. (2025). Crc4ever: Cyclic redundancy check for enhanced verification and efficient routing. In *Proceedings of the ACM SIGCOMM 2025 Posters and Demos*, ACM SIGCOMM Posters and Demos '25, page 178–180, New York, NY, USA. Association for Computing Machinery.

Borges, E. S., Bonella, V. B., Dos Santos, A. J., Menegueti, G. T., Dominicini, C. K., and Martinello, M. (2023). In-situ proof-of-transit for path-aware programmable networks. In *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*, pages 170–177. IEEE.

Borges, E. S., da Cunha Pontes, E., Mate, C., Loui, F., Martinello, M., and Ribeiro, M. R. (2022). Freerouter in a nutshell: A"protocoland"routing platform for open and portable carrier-class testbeds. In *Workshop de Testbeds*, pages 36–46. SBC.

Borges, E. S. et al. (2024a). Pint-box: Path-aware networking in a tofino box. In *2024 IEEE NFV-SDN*, pages 1–2.

Borges, E. S., Martinello, M., Bonella, V. B., dos Santos, A. J., Gomes, R. L., Dominicini, C. K., Guimarães, R. S., Menegueti, G. T., Barcellos, M., and Ruffini, M. (2024b). Pot-polka: Let the edge control the proof-of-transit in path-aware networks. *IEEE Transactions on Network and Service Management*.

Bu, K., Laird, A., Yang, Y., Cheng, L., Luo, J., Li, Y., and Ren, K. (2020). Unveiling the mystery of internet packet forwarding: A survey of network path validation. *ACM Computing Surveys (CSUR)*, 53(5):1–34.

Dominicini, C. et al. (2020). Polka: Polynomial key-based architecture for source routing in network fabrics. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, pages 326–334. IEEE.

Godfrey, P. B., Ganichev, I., Shenker, S., and Stoica, I. (2009). Pathlet routing. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, SIGCOMM '09, page 111–122, New York, NY, USA. Association for Computing Machinery.

Jia, Z., Zhang, Z., Rexford, J., Walker, D., and Wetherall, D. (2020). Adding path awareness to the internet architecture. *IEEE Transactions on Network and Service Management*, 17(3):2080–2093.

Krähenbühl, C., Wyss, M., Basin, D., Lenders, V., Perrig, A., and Strohmeier, M. (2023). FABRID: Flexible Attestation-Based routing for Inter-Domain networks. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5755–5772, Anaheim, CA. USENIX Association.

Legner, M., Klenze, T., Wyss, M., Sprenger, C., and Perrig, A. (2020). Epic: every packet is checked in the data plane of a path-aware internet. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA. USENIX Association.

Martinello, M., Gomes, R. L., Borges, E. S., Layber, H. C., Bonella, V. B., Dominicini, C. K., Guimarães, R., Ribeiro, M., and Barcellos, M. (2024). Pathsec: Path-aware secure routing with native path verification and auditability. In *2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pages 1–7.

Naous, J., Walfish, M., Nicolosi, A., Mazieres, D., Miller, M., and Seehra, A. (2011). Verifying and enforcing network paths with icing. In *Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies*, pages 1–12.

Postel, J. (1981). Internet protocol. RFC 791, Internet Engineering Task Force (IETF).

Sunshine, C. A. (1977). Source routing in computer networks. *ACM SIGCOMM Computer Communication Review*, 7(1):29–33.

Szabo, N. S. and Tanaka, R. I. (1967). *Residue Arithmetic and Its Applications to Computer Technology*. McGraw-Hill, New York.

Trammell, B. (2022). Current open questions in path-aware networking. *IRTF, RFC 9217*.

Ventorim, D., Borges, E., Guimarães, R., Martinello, M., Ribeiro, M., Dominicini, C., Schwarz, M., Xavier, B., Bezerra, J., Kiran, M., and Newman, H. (2025). A path-aware routing for data intensive science: Proposal, deployment and evaluation in high-performance testbed. In *Anais do XXX Workshop de Gerência e Operação de Redes e Serviços*, pages 71–84, Porto Alegre, RS, Brasil. SBC.

Wu, B. et al. (2018). Enabling efficient source and path verification via probabilistic packet marking. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pages 1–10.

Zhang, F. et al. (2014). Mechanized network origin and path authenticity proofs. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 346–357, New York, NY, USA. ACM.