Advancing Resilient Communication in Industrial Networks with P4-Programmable Data Planes and Multipath Transport Protocols

Ina Berenice Fink, Aurel Weinhold, and Klaus Wehrle
Communication and Distributed Systems, RWTH Aachen University, Germany
{fink, weinhold, wehrle}@comsys.rwth-aachen.de

***Abstract*—Industrial networks, and especially critical infrastructure, require highly reliable and low-latency communication to ensure continuous operation. Traditional redundancy mechanisms such as Highly Available Seamless Redundancy (HSR), Parallel Redundancy Protocol (PRP), and Multiprotocol Label Switching Transport Profile (MPLS-TP) provide deterministic protection but rely on static configurations and dedicated hardware, limiting flexibility and fallback options. At the same time, the increasing interconnection of previously isolated networks introduces additional points of failure that need to be covered by resilience mechanisms. To address these challenges, we present RedPath, a framework that combines P4-based network-level path control with multipath transport protocols such as Multipath TCP (MPTCP) to enable deterministic transmission over disjoint paths within a network, while supporting the use of wireless backup links. Our initial proof of concept of the path control, implemented using BMv2 software switches, demonstrates stable operation and fast failover times in the range of milliseconds, indicating its feasibility. Future work will validate the approach on programmable hardware.**

***Index Terms*—Telecommunication network reliability, transport protocols, critical infrastructure, redundancy**

I. Introduction

Ensuring reliable and resilient communication within industrial and critical infrastructures, e.g., energy grids, is essential to sustain continuous operation [1], [2] and prevent severe human and economic consequences. However, increasing cyberattacks and climate-related hazards, such as floods and wildfires, pose significant threats, while the growing interconnection of previously isolated systems introduces additional points of failure that must be considered in resilience concepts. To this end, wireless backup links, e.g., via cellular networks, are increasingly integrated into communication architectures to complement wired connections and maintain service continuity during large-scale outages [3], [4]. Nevertheless, proprietary fiber backbones remain the foundation of critical infrastructures, as cellular systems cannot yet meet the ultra-low latency demands of real-time control traffic [2], and the inherent variability of wireless channels challenges their reliability [5]. Consequently, resilience mechanisms must support both redundant communication within the wired backbone network *and* seamless failover to wireless backup links. In addition, strict latency requirements of time-critical applications impose constraints on failover mechanisms and path selection.

Popular solutions such as MPLS-TP [6], which provides deterministic redundant communication within the same network, do not natively support the integration of wireless backup links or the flexible use of more than two redundant paths. Recent work proposes to exploit multipath transport protocols to allow end devices redundant communication over multiple interfaces and heterogeneous networks [7]. For instance, MPTCP [8] enables the transparent use of multiple paths for multi- and single-homed devices, and its v0 implementation supports redundant transmission [9]. However, MPTCP alone cannot enforce disjoint redundant paths within the same network due to router-controlled path selection, and thus requires additional mechanisms to control intra-network path diversity.

To address these limitations, this paper proposes RedPath: a framework that combines multipath transport protocols at end devices with Software-Defined Networking (SDN)-based path control using P4-programmable data planes. This combination enables both flexible communication across heterogeneous networks and explicit control over path selection within the backbone network. SDN offers a global network view and, when paired with P4-programmable data planes, enables custom packet processing at line-rate and fast reconfiguration in response to network changes [3], allowing deterministic yet flexible use of redundant paths within networks. At the same time, multipath transport protocols enable flexible end-host control of redundant transmissions without application modifications [2], facilitating integration into existing infrastructure. In this work, we focus on SDN-based path control and demonstrate how programmable data planes enable efficient and scalable enforcement of disjoint paths.

Contributions. Our main contributions are as follows.

- We present a framework for redundant communication in heterogeneous networks using multipath transport protocols and SDN, including the design of a P4-based data plane for enforcing the distribution of subflows over disjoint paths in the same network.
- We implement a proof-of-concept using BMv2 software switches to demonstrate feasibility, as a first step toward deployment on programmable hardware.
- We evaluate our proof-of-concept using emulated networks with different topologies, assessing its performance and scalability.

Open Science Statement. We open-source the code of our framework [10] under the GPLv3 license.

© IEEE, 2026. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. DOI: TBD

II. Towards Flexible and Scalable Resilience in Industrial Networks

We discuss traditional approaches to resilient communication in today's industrial networks, then introduce multipath transport protocols and derive requirements for future solutions.

A. Industrial Networks and Redundancy Mechanisms

Characteristics of Modern Industrial Networks. Networks in industrial and critical infrastructure are becoming increasingly interconnected to support automated decision-making, and remote monitoring and control. As a result, industrial communication networks now range from local deployments within single plants to wide-area networks (WANs) spanning hundreds of kilometers [11], [12]. To meet the stringent latency and reliability requirements of time-critical applications, these networks often rely on dedicated fiber-optic infrastructure and deterministic communication technologies [13], [14]. At the same time, modern industrial networks are evolving towards more heterogeneous architectures, for example by integrating wireless links [4], [13] to reduce costs or add redundancy. This evolution increases the availability of redundant paths and makes resilience increasingly dependent on how efficiently the given path diversity can be utilized. Existing redundancy mechanisms, however, were largely designed for static and homogeneous environments and therefore provide limited flexibility as discussed below.

Traditional Redundancy Mechanisms. The HSR [15] and PRP [16] standards provide seamless recovery after link failures, unlike protocols such as RSTP [17], which require network reconfiguration, introducing additional delay. However, HSR and PRP require protocol-specific Ethernet frame handling (e.g., of HSR tags or PRP trailers) and impose topology or infrastructure constraints including specialized hardware, which can increase deployment costs and typically limits their deployment to dedicated Ethernet segments. Therefore, in wide-area deployments, operators commonly adopt MPLS-TP [6], [18], which offers SONET/SDH-like OAM, fast reroute, and path protection via predefined backups. While MPLS-TP targets recovery within 50 ms, fault detection and failover still cause non-negligible delays. Consequently, ultra-low latency requirements below 10 ms (for instance, in teleprotection [19]) cannot always be guaranteed and, instead, require simultaneous redundant transmission over disjoint paths. Although MPLS-TP supports *1+1 protection*, which sends traffic redundantly over a working and a protection path, redundancy is typically limited to a single statically configured backup path. Therefore, with traditional protection mechanisms, available path diversity (and especially wireless backup links) may remain underutilized, potentially limiting resilience, and motivating approaches that enable the coordinated and flexible use of redundant paths in heterogeneous IP-based networks. A natural way to exploit such path diversity is provided by multipath transport protocols.

B. Multipath Transport Protocols

Multipath transport protocols provide a flexible means to simultaneously exploit different network paths, while operating transparently to applications and allowing per-connection control. Two major protocols are MPTCP [8] and Multipath-QUIC (MPQUIC) [20] presented below.

MPTCP. MPTCP allows multiple subflows within a single TCP connection, and thereby supports the use of different interfaces (e.g., wired and wireless) or ports for the same connection. When one endpoint lacks MPTCP support, the connection falls back to standard TCP, ensuring backward compatibility. Subflows and connections are identified by tokens exchanged during the initial three-way handshake, and packet schedulers determine how traffic is distributed among subflows. Path managers additionally define which interfaces to use. Each subflow terminates with the regular TCP FIN/ACK exchange, whereas a DATA_FIN marks the end of the overall MPTCP connection. While a redundant scheduler is part of MPTCP's out-of-tree implementation [21], recent research proves its potential for performant failovers with multi-homed and mission-specific end devices [2].

MPQUIC. MPQUIC extends QUIC to support multiple paths per connection. Since QUIC natively multiplexes multiple streams and embeds a connection ID (CID) in each packet header, mapping packets to flows is straightforward. However, all payload and most header fields, including CIDs and CONNECTION_CLOSE signals, are encrypted, preventing inspection or modification by middleboxes. Furthermore, MPQUIC remains at a prototype stage [22] and currently lacks redundancy support.

To conclude, multipath transport protocols, and especially MPTCP, offer a promising alternative for achieving communication resilience without specialized hardware or the rigidity of traditional protection mechanisms (cf. Sec. II-A). However, due to their operation at end devices, multipath transport protocols can only choose which interfaces and how many subflows to use, but not which specific paths to take within a network. Additional mechanisms are therefore required to enable coordinated and deterministic use of path diversity.

C. Requirements for Resilience in Industrial Networks

Given the limitations of existing resilience mechanisms and the performance and availability requirements of industrial and critical infrastructures, novel resilience solutions should fulfill the following requirements:

- **R1 (Path Diversity):** Deterministic distribution of redundant data streams across multiple disjoint paths with seamless integration of heterogeneous backup technologies (e.g., wireless links).
- **R2 (Operational Performance):** Scalable operation across large network topologies while maintaining low failover latency to meet industrial timing requirements.
- **R3 (Deployability):** Compatibility with existing IP-based industrial networks without requiring specialized redundancy hardware or application changes.

Coordinated control is well supported by SDN-based architectures, which enable forwarding decisions to be computed centrally. P4-programmable data planes complement this capability by enabling fine-grained and flexible packet processing directly within network devices at line rate. Together, SDN and P4 therefore provide a promising foundation for meeting the requirements outlined above. Building on these capabilities, we propose RedPath, which combines P4-programmable switches with multipath transport protocols. Before presenting RedPath, we first review related work addressing redundant transmission in industrial networks in the next section.

Takeaway: *Existing redundancy mechanisms lack flexibility anddo not exploit path diversity in modern networks, motivating approaches for coordinated multipath utilization.*

III. Related Work on Resilient Communication

Prior work has explored redundancy mechanisms at different layers of the network stack. At the network layer, researchers adapted PRP to operate in IP-based networks [23], [24], but these approaches do not consider heterogeneous networks (R1ϟ). Li et al. [25] propose a P4-based solution for packet replication, forwarding over disjoint paths, and duplicate elimination at the egress switch. However, heterogeneous backup links are again not considered (R1ϟ). In addition, duplicate handling is performed by the P4 switches, requiring per-packet state in the data plane and increasing resource consumption (R2ϟ). At the transport layer, Felix et al. [26] extended MPTCP to enable redundant transmission in wireless networks, but their work does not allow the use of multiple disjoint paths within wired networks (R1ϟ). Du et al. [27] use MPTCP with OpenFlow to distribute subflows across disjoint paths in satellite networks. However, their approach assumes multi-homed devices and is therefore also not applicable to enable redundant communication within the same network (R1ϟ) while lacking the flexibility and performance of P4-programmable data planes (R3ϟ). Last, Lopez et al. [9] combine MPTCP with MPLS to provide redundancy for rail signaling where MPLS routers take care of distributing subflows over disjoint paths. While their approach integrates well with existing MPLS infrastructures and potentially allows the use of wireless backup links at end devices, Lopez et al. point out that using MPLS suffers from limited flexibility and scalability due to manual route calculation and configuration (R3ϟ).

Overall, most approaches treat heterogeneous redundancy and intra-network path diversity separately, while combined solutions rely on static mechanisms such as MPLS, limiting flexibility and scalability. At the same time, prior work on fast path discovery [28], fast reroute [29], and load balancing [30]–[32] highlights the potential of SDN to enable dynamic multipath routing. Thus, the remainder of this paper investigates how multipath transport protocols can be combined with P4-programmable data planes to provide a flexible and programmable approach for exploiting path diversity across heterogeneous networks.

Takeaway: *Existing approaches fail to flexibly and scalably combine heterogeneous redundancy with controlled intra-network path diversity, motivating the need for programmable and dynamic solutions.*

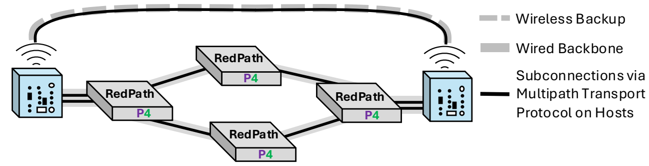


Fig. 1: End devices use wired and wireless links for redundant transmission via multipath transport protocols, while RedPath enforces the disjoint distribution of subconnections in the backbone network using P4-programmable switches.

IV. P4-Enabled Redundant Multipath Transmission in Heterogeneous Networks

In this section, we present RedPath, a flexible, and performant framework for redundant heterogeneous communication in industrial (wide area) networks. We provide an overview of RedPath, before detailing its workflow with a focus on programmable path control.

A. Overview

RedPath enables redundant communication across heterogeneous network infrastructures by combining multipath transport protocols at end devices with P4-programmable switches in the backbone network (cf. Fig. 1). Using multipath transport protocols, end devices can simultaneously utilize heterogeneous access technologies (e.g., wired and wireless links) and generate redundant subflows while P4-programmable switches enable explicit control over their distribution across disjoint backbone paths (R1✓). In this work, we focus on the latter, i.e., the design of a scalable and programmable path control mechanism in the data plane. To this end, RedPath separates centralized path computation from data-plane packet processing, enabling scalable path management while keeping packet processing solely in the data plane to meet the strict timing requirements of mission-critical traffic (R2✓). However, in contrast to P4-based related work [25], RedPath shifts redundancy generation to end devices rather than duplicating packets within the network, reducing data-plane overhead. For the same reason, RedPath also operates transparently to applications and does not require dedicated redundancy hardware or modifications to industry-specific end devices, higher-layer protocols, or applications, enabling incremental deployment in existing industrial infrastructures (R3✓). In addition, RedPath's design allows the integration of P4-based in-network telemetry applications to enable timely detection of and reaction to network changes, further strengthening the reliability of redundant paths and decreasing overhead by selectively activating paths.

B. P4-based Path Calculation and Connection Management

In the following, we detail RedPath's path calculation and connection management specifically designed for P4-programmable network devices. As illustrated in Fig. 2, the control plane ① calculates and installs the disjoint paths; the data plane ② identifies multipath connections and subconnections, ③ registers new (sub)connections and path assignments, and ④ forwards packets while keeping track of the connection states. Although MPQUIC does not yet support redundant

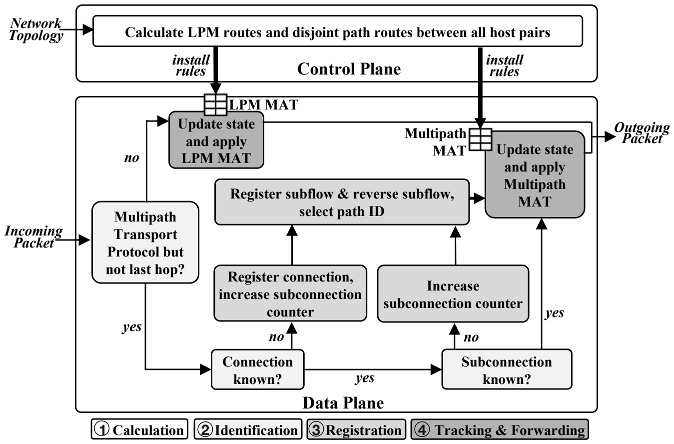


Fig. 2: RedPath's workflow: ① forwarding rules in the control plane for enforcement in the data plane using Match-Action Tables (MATs), ② identify multipath packets, ③ register unknown (sub)connections and assign paths, and ④ update (sub)connection states and forward packets.

transmission, it remains a relevant future alternative to MPTCP due to its reported performance advantages [33]. We therefore consider both MPTCP and MPQUIC in the design of our approach and describe how the proposed P4-based resilience mechanism can be applied to both protocols. Our practical implementation and evaluation, however, focus on MPTCP. In the following, we use the term *subconnection* to collectively refer to MPTCP subflows and MPQUIC substreams. Packets using a multipath transport protocol are called *multipath packets*; all others are *non-multipath packets*.

① Path Calculation. Due to our use case, i.e., proprietary industrial networks, we assume that the network topology is managed and known by the operator of our framework, and can serve as input for the control plane to compute disjoint paths through the network. Each path is assigned a unique path ID, and each subconnection is assigned to one of the available paths through its ID. Using match-action tables, the control plane enforces the computed paths and assignments in the switches' data planes as follows: for each switch and path, it extracts the egress port corresponding to the path's next hop. Then, it installs an action that matches to the assigned path of a subconnection and triggers forwarding over the respective egress port. To handle the last hop and non-multipath packets, the control plane installs Longest Prefix Match (LPM)-routes in an additional match-action table, which matches against destination IP addresses to derive the egress port. Due to its global view, the control plane can recalculate paths in response to topology or performance changes, e.g., detected through P4-based in-network telemetry applications running on the same switches, and installs updated forwarding rules.

② Identification of (sub)connections. For MPTCP, multipath packets can be identified by the presence of the MPTCP option in the TCP header. For MPQUIC, RedPath exploits CID prefixes, i.e., if two flows use different UDP ports but show same CID prefix, they are registered as subconnections of the same multipath session. To keep track of multipath connections, the data plane maintains a register-based hash map that uses the hash of a connection's source and destination IP address. Optionally, it also stores additional identifiers such as the sender token in the case of MPTCP and the CID in the case of MPQUIC to support and distinguish multiple connections between the same pair of hosts. The register value is initially set to 0 and changed to 1 when a new connection is first detected. Although MPTCP's sender token is derived using SHA-256, which cannot be computed in current P4-programmable data planes, the data plane can send a copy of the first packet of a new connection to the control plane, which computes the token and returns it. The data plane then stores the resulting hash and updates the relevant registers. Although this operation introduces additional processing delay, it is performed only once during connection establishment, before regular data transmission begins. Subsequent packets carry the token in plaintext and therefore do not incur this delay. Since the delay occurs only at the beginning of a connection rather than during ongoing transmission, it is expected to be acceptable even in industrial environments. If only a single multipath connection exists between two hosts or if each interface has a unique IP address, the additional identifiers are not required and can be omitted. Moreover, future P4-programmable data planes may provide support for more sophisticated hash computations. Subconnections of both protocols are distinguished by their transport layer source and destination ports, and tracked using separate registers.

③ (Sub)Connection registration and path assignment. RedPath maintains counters for active subconnections and available paths to ensure balanced distribution. When a new multipath connection is detected, its register entry and subconnection counter are initialized with 1, and the first subconnection is assigned to the path with ID = 1. If the connection already exists but a new subconnection is detected, the counter is incremented, and its path ID is assigned as follows: if the number of subconnections does not exceed the number of available paths, the next free path ID is selected. Otherwise, a path is chosen within the available ID range using round-robin or randomization. For each new subconnection, RedPath also registers the reverse subconnection, assigning it the ingress port of the current packet as egress port to ensure that packets traverse symmetrical disjoint paths in both directions. If the destination is a single hop away, the connection is not registered to save storage; only its reverse subconnection is maintained.

④ Forwarding and state tracking. Non-multipath packets and those with only a single remainig hop are forwarded using LPM-based forwarding. Multipath packets, by contrast, are forwarded according to their assigned path ID or, for reverse subconnections, using the stored egress port. Two match-action tables implement this logic, rewriting the packet's egress port before transmission. When both directions of a MPTCP subconnection have exchanged FIN flags, RedPath marks the subconnection as closed and resets its register entry. The connection's counter is decremented, and when it reaches zero, the entire connection entry is cleared. To keep track of subconnection states in case of MPTCP, i.e., storing observed flags, additional registers are used and updated for each packet. MPQUIC subconnections are excluded from this due to their encryption. Here, connection termination can be detected through timeouts (cf. Sec. II-B).

We envision deploying RedPath on P4-programmable hardware targets to enable line-rate packet processing in future deployments. However, since this paper focuses on the general design and feasibility of the proposed approach, a hardware implementation is beyond its scope. As a first step, we therefore implemented and evaluated a software-based proof of concept using BMv2 and Mininet.

Takeaway: *Based on P4, RedPath enforces the performant and disjunct distribution of redundant transmissions in wired (backbone) networks, while end devices flexibly use heterogeneous links via multipath transport protocols.*

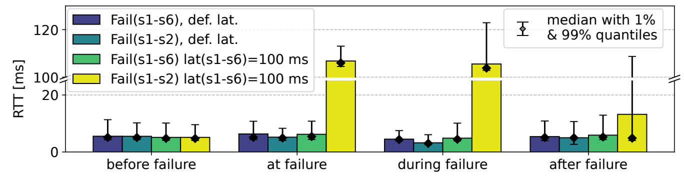
V. Proof-of-Concept and Evaluation

We evaluate RedPath using Mininet [34] to assess its feasibility and system-level behavior prior to a future hardware implementation. Our evaluation focuses on the use of disjoint paths within the same network, i.e., the SDN-based path control component of RedPath. The use of multipath transport protocols for redundant transmission across heterogeneous networks has been sufficiently demonstrated in prior work [2] and is therefore not considered. Mininet allows us to explore larger and more diverse topologies than would be feasible with hardware resources. While open-source models of hardware switches, such as Intel Tofino, are promising, their integration into reproducible Mininet workflows is still evolving. We therefore opted for the BMv2 software switch model [35], a popular platform for prototyping and testing P4 programs. We evaluate RedPath's feasibility, performance, and control-plane overhead using different topologies and path configurations, then analyze its resource efficiency and scalability. Last, we discuss benefits, limitations, and future work.

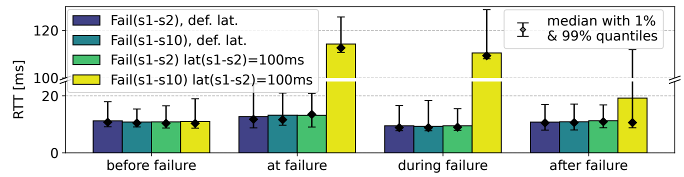
A. Proof-of-Concept

We implemented the data plane of RedPath in P4 targeting BMv2, while the control plane was implemented in Python3. Using Mininet, we deployed it in an emulated environment with multiple switches and hosts, and we ran MPTCPv0 [21] on the hosts to create redundant subflows (cf. Sec. II-B), considering a single MPTCP connection between two endpoints to isolate and evaluate the path control mechanism. We further adapted the *DIRECT* algorithm [36] to compute the shortest and the $k-1$ near-shortest disjoint paths between all edge nodes, such that every subconnection was assigned one of the paths.

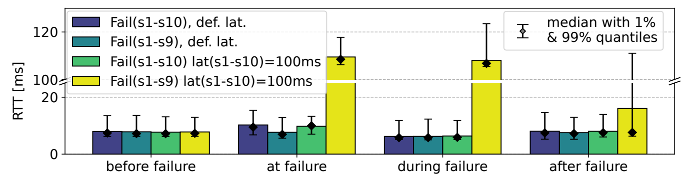
Feasibility Test Setup. To assess RedPath's functionality and scalability, we tested ring, full-mesh, and random topologies of varying sizes up to 50 switches, with two attached hosts. The maximum topology size was determined by the computational resources required to emulate each switch in software using BMv2 and Mininet. Larger topologies exceeded the available resources of the evaluation system. However, this limitation concerns only the emulation environment and does not impose an inherent limit on the network size supported by our design.



(a) Full mesh



(b) Ring



(c) Random with 30% link probability between two nodes

Fig. 3: RTTs before, at, during, and after emulating an outage of one link for different topologies with 10 switches.

For each topology, RedPath computed between two and five disjoint paths between both hosts, installing the corresponding forwarding rules in the switches' data planes. On the transport layer, MPTCP was configured to open the same number of subflows as available disjoint paths, using identical IP addresses but distinct TCP ports. A TCP client then transmitted packets from one host to a TCP server on the other host at regular intervals. Each packet was echoed back to the sender to measure its Round-Trip Time (RTT). We captured all traffic at both hosts and intermediate switches to verify path diversity and observe RedPath's reaction to link failures, which we triggered for a duration of 10 s at random points in time after transmission start. Each test ran for 60 s and was repeated 30 times.

Observations. Across all topologies, the destination-side switch received traffic over all configured subflows across distinct paths. When a link failed, the corresponding subflows were temporarily lost, but the overall connection remained stable, and data continued to be transmitted via the remaining subflows, proving the feasibility of our design. Exact failover times are provided below.

B. Failover Performance

We measured the RTT before, during, and after link failures over 30 runs for each topology with 10 switches and two subflows, using the experimental setup described above. Two scenarios were considered: a) uniform latency, i.e., all links had a default propagation delay of a few milliseconds, and b) asymmetric latency, i.e., one link was configured with a 100 ms delay to emulate path heterogeneity.

Results. We present the results in Fig. 3a–3c. Across all topologies, RedPath maintained session continuity with failover delays in the order of 1 ms to 3 ms, confirming its robustness and topology independence. While the RTT of the first packet affected by a failure exhibited higher variability, this

TABLE I: Combined computation and installation time of disjoint paths versus topology (# nodes, # calc. paths).

Ring(2,2)	Ring(10,2)	Ring(30,2)	Ring(50,2)	Random(50,2)	Random(50,5)
2.076 s	2.78 s	4.82 s	6.87 s	6.89 s	7.13 s

likely resulted from the limited number of samples for this event. Furthermore, in the uniform-latency scenario, the RTT decreased slightly during link failures, regardless of which link failed. We attribute this RTT decrease to reduced emulation load when fewer packets were processed concurrently. Notably, when a higher-latency link (100 ms) failed, the RTT remained low, as packets continued over the unaffected, low-latency paths. Conversely, when the low-latency path failed, the RTT increased to approximately 100 ms for the failure duration, as expected. Last, the RTT remained increased for several seconds after link restoration. Here, packet traces revealed that MPTCP required time to reintegrate the recovered path, consistent with prior observations in [2]. We expect hardware deployments to yield lower and more consistent failover times, which we will validate in the future.

C. Control-Plane Overhead

To evaluate control-plane responsiveness during full network configuration, we measured the time required to recompute paths and install forwarding rules across the entire network. Such a full configuration is performed only during the initial setup of RedPath or after network-wide topology changes. For each topology, RedPath computed and installed between two and five disjoint paths between all edge nodes.

Results. Path computation and rule installation required at least 2 s for minimal topologies and up to 7.13 s for networks with 50 switches, including ring and random topologies with a link probability of 30 % (cf. Tab. I). The total duration varied only slightly with the number of paths computed. Profiling with py-spy [37] showed that path computation and rule installation each accounted for approximately 50 % of the total duration. Consequently, replacing all forwarding rules in a 50-switch network would cause a temporary interruption of approximately 3.5 s. During normal operation, expected isolated link or switch failures are handled by the precomputed redundant paths and therefore do not trigger control-plane involvement.

D. Resource Efficiency and Scalability

To assess the feasibility of deploying RedPath in industrial networks, we analyze the scaling behavior of its data-plane state requirements. Since BMv2 abstracts from hardware-specific TCAM and SRAM capacities, we evaluate the asymptotic state complexity of our design. RedPath maintains state per subflow rather than per packet, resulting in O(connections) state in the best case and O(connections×subflows) in the worst case. In contrast, redundancy approaches based on packet replication require maintaining per-packet state [25], significantly increasing memory pressure on switches. Therefore, by shifting redundancy to the transport layer, RedPath preserves scarce hardware resources, allowing to cover a larger number of connections. This also facilites the execution of multiple concurrent applications per switch (e.g., telemetry, monitoring, or traffic engineering), allowing sustainable and effective hardware utilization.

E. Benefits, Limitations, and Future Work

As demonstrated by our proof-of-concept implementation and evaluation, RedPath enables the efficient and scalable use of multiple disjoint paths within a single network, achieving low failover times suitable for industrial communication. The integration of multipath transport protocols at the end devices inherently enables extension to heterogeneous wired and wireless links. RedPath's design is also compatible with other P4-based applications, such as telemetry systems, enabling the automated detection of and reaction to network changes (e.g., [38]), which we will explore in future work. Our current implementation focuses on MPTCP with a single connection between two endpoints. Future work will extend RedPath to multiple concurrent connections and to MPQUIC, broadening its applicability. One important initial step toward this is the extension of MPQUIC to support redundant transmission. Future work should also consider the implementation of the control plane, as well as its susceptibility and resilience to network failures. Furthermore, the feasibility of dynamic reconfiguration depends on the latency requirements of the respective application and should therefore be evaluated across different application scenarios in industrial networks. Finally, we plan to validate our approach on hardware targets, which are expected to lower failover times significantly.

Takeaway: *Our evaluation shows that RedPath enables transport-layer-based redundancy with low failover times across diverse topologies while reducing data-plane resource requirements, motivating future hardware validation.*

VI. Conclusion

In this paper, we present RedPath, a framework that combines SDN-based path control using P4-programmable switches with multipath transport protocols at end devices to enable redundant transmission in heterogeneous networks with wired and wireless links. By enabling explicit path selection in the data plane and multipath management at the transport layer, RedPath improves flexibility and allows efficient utilization of path diversity compared to traditional redundancy solutions. Our proof-of-concept implementation on BMv2 software switches demonstrates the feasibility and effectiveness of the proposed path control approach. In future work, we plan to port RedPath to programmable hardware such as Intel Tofino, integrate telemetry feedback, and evaluate control-plane resilience under performance degradation and topology changes in industrial real-time networks.

Acknowledgments

Funded by the German Federal Ministry for Economic Affairs and Energy of Germany (BMWE) — Research Project VeN2uS — 03EI6053K.

REFERENCES

[1] M. Shahraeini and P. Kotzanikolaou, *Resilience in Wide Area Monitoring Systems for Smart Grids*, pp. 555–569. Cham: Springer International Publishing, 2021.

[2] I. B. Fink, L. Ferlemann, M. Dahlmanns, C. Thimm, and K. Wehrle, "Emulating and Evaluating Transport Layer Protocols for Resilient Communication in Smart Grids," in *NOMS 2025-2025 IEEE Network Operations and Management Symposium*, pp. 01–10, 2025.

[3] A. Aydeger, N. Saputro, K. Akkaya, and S. Uluagac, "SDN-enabled recovery for Smart Grid teleprotection applications in post-disaster scenarios," *Journal of Network and Computer Applications*, vol. 138, pp. 39–50, 2019.

[4] M. Lorenz, T. M. Pletzer, M. Schuhmacher, T. Sowa, M. Dahms, S. Stock, D. Babazadeh, C. Becker, J. Jaeger, T. Lorz, M. Dahlmanns, I. B. Fink, K. Wehrle, A. Ulbig, P. Linnartz, A. Selimaj, and T. Offergeld, "Interconnected network protection systems - the basis for the reliable and safe operation of distribution grids with a high penetration of renewable energies and electric vehicles," in *CIRED Porto Workshop 2022: E-mobility and power distribution systems*, vol. 2022, pp. 548–552, 2022.

[5] J. Bodenhausen, C. Sorgatz, T. Vogt, K. Grafflage, S. Rötzel, M. Rademacher, and M. Henze, "Securing Wireless Communication in Critical Infrastructure: Challenges and Opportunities," in *Mobile and Ubiquitous Systems: Computing, Networking and Services* (A. Zaslavsky, Z. Ning, V. Kalogeraki, D. Georgakopoulos, and P. K. Chrysanthis, eds.), (Cham), pp. 333–352, Springer Nature Switzerland, 2024.

[6] D. Brungard, M. Betts, S. Ueno, B. Niven-Jenkins, and N. Sprecher, "Requirements of an MPLS Transport Profile." RFC 5654, Sept. 2009.

[7] M. Dahlmanns, I. B. Fink, G. Erichsen, G. Lin, T. Hammer, B. Borkenhagen, S. Schneider, C. Maahsen, and K. Wehrle, "Reliable and secure control center to station device communication," in *CIRED 2024 Vienna Workshop*, vol. 2024, pp. 540–543, 2024.

[8] A. Ford, C. Raiciu, M. J. Handley, O. Bonaventure, and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses." RFC 8684, Mar. 2020.

[9] I. Lopez, M. Aguado, D. Ugarte, A. Mendiola, and M. Higuero, "Exploiting redundancy and path diversity for railway signalling resiliency.," in *2016 IEEE International Conference on Intelligent Rail Transportation (ICIRT)*, pp. 432–439, 2016.

[10] I. B. Fink, A. Weinhold, and K. Wehrle, "BMv2 Implementation of RedPath." https://github.com/COMSYS/RedPath, 2026.

[11] E. D. Knapp, "4 - Introduction to Industrial Control Systems and Operations," in *Industrial Network Security* (E. D. Knapp, ed.), pp. 65–90, Syngress, Third Edition ed., 2024.

[12] E. D. Knapp, "5 - Industrial Network Design and Architecture," in *Industrial Network Security* (E. D. Knapp, ed.), pp. 91–128, Syngress, Third Edition ed., 2024.

[13] B. Achaal, M. Adda, M. Berger, H. Ibrahim, and A. Awde, "Study of smart grid cyber-security, examining architectures, communication networks, cyber-attacks, countermeasure techniques, and challenges," *Cybersecurity*, vol. 7, no. 1, p. 10, 2024.

[14] E. D. Knapp, "6 - Industrial Network Protocols," in *Industrial Network Security* (E. D. Knapp, ed.), pp. 129–179, Syngress, Third Edition ed., 2024.

[15] H. Kirrmann, K. Weber, O. Kleineberg, and H. Weibel, "Seamless and low-cost redundancy for substation automation systems (high availability seamless redundancy, HSR)," in *2011 IEEE Power and Energy Society General Meeting*, pp. 1–7, 2011.

[16] H. Kirrmann, M. Hansson, and P. Muri, "IEC 62439 PRP: Bumpless Recovery for Highly Available, Hard Real-Time Industrial Networks," in *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*, pp. 1396–1399, 2007.

[17] "IEEE Standard for Information Technology -Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Common Specifications - Part 3: Media Access Control (MAC) Bridges: Amendment 2 - Rapid Reconfiguration," *IEEE Std 802.1w-2001*, pp. 1–116, 2001.

[18] E. Gray, D. N. N. Bitar, L. Fang, L. Berger, and L. Andersson, "MPLS Transport Profile (MPLS-TP) Control Plane Framework." RFC 6373, Sept. 2011.

[19] L. F. F. De Almeida, J. R. D. Santos, L. A. M. Pereira, A. C. Sodré, L. L. Mendes, J. J. P. C. Rodrigues, R. A. L. Rabelo, and A. M. Alberti, "Control Networks and Smart Grid Teleprotection: Key Aspects, Technologies, Protocols, and Case-Studies," *IEEE Access*, vol. 8, pp. 174049–174079, 2020.

[20] Y. Liu, Y. Ma, Q. D. Coninck, O. Bonaventure, C. Huitema, and M. Kühlewind, "Multipath Extension for QUIC," Internet-Draft draft-ietf-quic-multipath-07, Internet Engineering Task Force, Apr. 2024. Work in Progress.

[21] C. Paasch, S. Barre, et al., "Multipath TCP in the Linux Kernel, available from https://www.multipath-tcp.org." https://www.multipath-tcp.org.

[22] "MPQUIC Prototype based on quic-go." https://github.com/qdeconinck/mp-quic.

[23] M. Rentschler and H. Heine, "The Parallel Redundancy Protocol for industrial IP networks," in *2013 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1404–1409, 2013.

[24] M. Popovic, M. Mohiuddin, D.-C. Tomozei, and J.-Y. L. Boudec, "iPRP—The Parallel Redundancy Protocol for IP Networks: Protocol Design and Operation," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1842–1854, 2016.

[25] Z. Li, L. Tian, J. Xia, and Y. Hu, "A Multipath Routing Mechanism by Packet Replication and Elimination for Reliable Communications.," in *ICCT*, pp. 1364–1369, IEEE, 2023.

[26] B. Felix, I. Steuck, A. Santos, S. Secci, and M. Nogueira, "Redundant Packet Scheduling by Uncorrelated Paths in Heterogeneous Wireless Networks," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 00498–00503, 2018.

[27] P. Du, S. Nazari, J. Mena, R. Fan, M. Gerla, and R. Gupta, "Multipath TCP in SDN-enabled LEO satellite networks," in *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pp. 354–359, 2016.

[28] T. Duan and V. Dinavahi, "Fast Path Recovery for Single Link Failure in SDN-Enabled Wide Area Measurement System," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1645–1653, 2022.

[29] D. De Lima, F. Vogt, A. T. da Silva, F. R. Cesen, and C. E. Rothenberg, "RESISTING: A New Fast-Reroute Mechanism with Packet Distribution on P4-Programmable Switches," in *2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 1–6, 2024.

[30] L.-Q. He and L. Ou, "Research on Programmable Data Plane Load Balancing based on Multipath Selection," in *Proceedings of the 3rd Annual International Conference on Electronics, Electrical Engineering and Information Science (EEEIS 2017)*, pp. 260–268, Atlantis Press, 2017/09.

[31] Y.-S. Chen, W.-Y. Chen, C.-H. Ke, and Y.-S. Yu, "Multipath Transmission Mechanism with P4 Switches in Software Defined Networks," in *2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, pp. 1–2, 2019.

[32] M. Pizzutti and A. E. Schaeffer-Filho, "Adaptive Multipath Routing based on Hybrid Data and Control Plane Operation," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 730–738, 2019.

[33] Q. De Coninck and O. Bonaventure, "Multipath QUIC: Design and Evaluation," in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '17, (New York, NY, USA), pp. 160–166, Association for Computing Machinery, 2017.

[34] "Mininet." http://mininet.org.

[35] "Behavioral Model (bmv2)." https://github.com/p4lang/behavioral-model.

[36] C. Häcker, P. Bouros, T. Chondrogiannis, and E. Althaus, "Most Diverse Near-Shortest Paths," in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '21, (New York, NY, USA), pp. 229–239, Association for Computing Machinery, 2021.

[37] "py-spy." https://github.com/benfred/py-spy.

[38] F. Alhamed, D. Scano, P. Castoldi, J. J. Vegas Olmos, I. Vershkov, F. Paolucci, and F. Cugini, "P4 Telemetry collector," *Computer Networks*, vol. 227, p. 109727, 2023.