Emulating and Evaluating Transport Layer Protocols for Resilient Communication in Smart Grids

Ina Berenice Fink, Lennart Ferlemann, Markus Dahlmanns, Christian Thimm, and Klaus Wehrle
Communication and Distributed Systems, RWTH Aachen University, Germany
{fink, ferlemann, dahlmanns, thimm, wehrle}@comsys.rwth-aachen.de

***Abstract*—The increasing integration of decentralized renewable energy resources and the drive for greater efficiency have accelerated the transition from traditional power grids to smart grids. This shift necessitates robust communication architectures to ensure grid stability and prevent blackouts. Fast and reliable communication is especially critical for exchanging measurements and configurations in adaptive grid protection systems, which must be resilient to link and device failures. Allowing the use of multiple communication paths within a single TCP connection, Multipath TCP (MPTCP)'s benefits have been well-researched in other domains but its potential for smart grids remains unexplored. In this paper, we address this gap by conducting a large-scale emulation of a real electric power distribution system's communication network, incorporating context-specific hardware. Our evaluation shows the feasibility and benefits of MPTCP for realizing failovers in smart grids compared to TCP and QUIC and explores the trade-offs of MPTCP's default and redundant schedulers in terms of usability and performance.**

***Index Terms*—Telecommunication network reliability, wide area networks, transport protocols, smart grids, redundancy**

I. Introduction

The reliable operation of smart grids depends heavily on interconnected communication infrastructure and comprehensive network management strategies. Key functions such as energy efficiency, grid protection and grid stability rely on continuous monitoring and control of the grid's state and device parameters [1], [2]. For example, the exchange of measurements and configuration parameters essential for adaptive grid protection necessitates dependable, grid-wide communication, both between the control center and substations as well as between substations themselves [3]–[5]. This communication must be resilient to link and on-path device failures, as these could otherwise cause malfunctions in grid protection, potentially leading to blackouts and brownouts. To mitigate such risks, hardware redundancies and efficient recovery mechanisms are critical for ensuring seamless grid operations [6], [7].

To address these challenges, separate backup channels – such as cellular or satellite communication – are an integral component of emerging smart grid architectures [3], [4], [7]. However, the efficacy of backup channels is highly dependent on their integration with appropriate communication protocols, which optimize failover processes and facilitate effective network management. In contrast, realizing failovers with single-path transport protocols like TCP and QUIC requires application reconfiguration, which may not be feasible with proprietary software, and results in significant delays caused by the need to re-establish connections and retransmit packets via the backup channel [8], [9].

IETF's Multipath TCP [10] offers a solution by enabling establishment of a single TCP connection over multiple paths, thus eliminating the need for connection re-establishment and providing a transparent failover process for applications. Furthermore, a redundant scheduler for MPTCP has been developed [9] [11], which enables simultaneous transmission over multiple channels and significantly reduces failover delays. However, MPTCP's usability with dedicated smart grid telecontrol hardware and the performance of its different schedulers in smart grid communication networks, compared to single-path protocols, remains unclear. This leaves grid operators with a complex choice of protocols and limited clarity regarding whether the performance demands of their applications can be met. Furthermore, authentic large-scale evaluations of communication protocols in smart grid Wide Area Networks (WANs) are constrained by the lack of testbeds that comprise real network topologies.

In this paper, we address these gaps by presenting a large-scale network emulation based on a real power grid communication network topology, along with a thorough evaluation of MPTCP's feasibility and performance for resilient inter-substation and substation-to-control-center communication. Our findings demonstrate that MPTCP can be effectively used with standard telecontrol hardware, and that only a combination of the redundant transmission with low-latency backup communication can meet the most stringent latency requirements of future smart grid applications.

Contributions. Our main contributions are as follows.

- We built an authentic large-scale network emulation based on the topology of a German Distribution System Operator (DSO), enabling realistic testing and evaluation of communication concepts and protocols.
- We assess the feasibility and complexity of using MPTCP with industry-specific telecontrol hardware.
- Using our emulation in conjunction with cellular backup communication, we conduct a real-world performance comparison between MPTCP, TCP, and QUIC, offering smart grid operators a comprehensive evaluation of these protocols in terms of behavior and performances.

Open Science Statement. We open-source the code for emulating our topology [12] under the GPLv3 license.

Paper Organization. In Sec. II, we introduce the architecture and requirements of distribution grid Information and Com-

munication Technology (ICT) infrastructure, review existing research on resilient smart grid communication, and derive open research questions. In Sec. III, we present the testbed developed to address these research questions. Sec. IV investigates the feasibility of deploying MPTCP on smart grid telecontrol hardware. In Sec. V, we use that hardware and the developed testbed to evaluate the failover performance of MPTCP, TCP, and QUIC. Based on these evaluations, Sec. VI assesses the suitability of these protocols for enhancing the resilience of smart grid communication. We conclude the paper in Sec. VII.

II. Towards Resilient Communication in Future Energy Grid Protection Systems

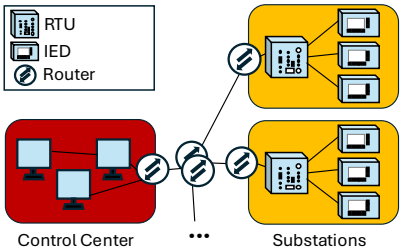
In this section, we first describe the architecture of modern distribution grid ICT infrastructure and outline its requirements. We then review recent research efforts and the potential of MPTCP to ensure resilience in smart grid communication.

A. Distribution Grid ICT Infrastructure

Distribution grid ICT infrastructures typically comprise a control center and distributed substations which accommodate various Intelligent Electronic Devices (IEDs), e.g., control and protection devices [13] (cf. Fig. 1a). Substations are equipped with Remote Terminal Units (RTUs) (cf. Fig. 1b) that play a critical role as they act as gateways for IEDs [14], enable communication between substations and the control center, and facilitate inter-substation communication. In essence, communication can be categorized into three subtypes [2]: a) control center to substation, b) inter-substation, and c) intra-substation communication. While each subtype entails distinct requirements necessitated by its application, we focus on a) and b) which share one WAN that allows the control center to centrally monitor and control the power grid, and enables direct information exchange between substations.

Basic Architecture and Performance Demands. WANs in distribution grid ICT infrastructure primarily consist of proprietary fiber optic networks [15] and increasingly rely on IP [16] to allow for scalable and flexible transmission [17], [18] in combination with MPLS technology to provide Quality of Service (QoS) guarantees [19], [20]. Specifically, operators are increasingly deploying MPLS-Transport Profile [21], which provides similar operations, administration and maintenance (OAM) functionality as SONET/SDH [20] and allows for fast failover in case of local failures within the WAN. Above, they implement TCP in the RTUs to provide reliable data transfer [22] in the WAN and industrial application layer protocols such as DNP3 or MMS [2], [15] that allow the RTUs to read and write values in the IEDs.

The exact performance demands vary for different functions but generally involve latencies ranging from a few milliseconds to a few seconds [2], [4], [23]. Most critically, tele-protection is said to require communication latencies below 10 ms [24]–[26]. To address these latency demands, different communication concepts and solutions have been proposed in related work.



(a) Distributed substations with Remote Terminal Units (RTUs) and Intelligent Electronic Systems (IEDs) connected to a control center.

(b) FW-5-GATE-4G [27] by LACROIX SAE IT-systems

Fig. 1: Basic overview a distribution system ICT infrastructure and an exemplary RTU device, also used in our evaluation.

B. Related Work on Resilient Smart Grid Communication

A substantial body of research has focused on communication within smart grid ICT infrastructure, particularly addressing performance requirements and proposing solutions to meet them (e.g., [2], [28], [29]). However, a growing area of interest is resilience, ensuring robust communication in the face of network failures or outages.

Resilience. Several studies have proposed architectures and techniques to improve network failover and path selection when redundant paths exist within the same network [30]–[32]. While these solutions offer improvements, they often fall short in mitigating the effects of large-scale outages of the core network, e.g., caused by severe weather conditions or targeted cyber-attacks. This affects especially rural areas where creating redundant wired connectivity is costly [33]. To address this issue, researchers have proposed hybrid communication architectures that incorporate multiple communication media and channels [3], [4], [7], e.g., providing wireless backup communication via cellular or satellite communications. However, the success of these architectures heavily depends on the used communication protocols and their ability to manage multiple channels effectively.

Multi-Channel Management. One approach to multi-channel management was proposed by Léon et al. [28] who introduce a multi-path and multi-channel routing protocol designed to enhance performance in wireless neighborhood area networks in smart grids. However, their work does not address wide-area communication or network failovers. Zhang et al. [34] propose a framework for resilient wide-area control in power grids using redundant communication channels, but their framework does not ensure compatibility with TCP/IP protocols. To enable transparent multipath management in IP-based grid protection systems, Lorenz et al. [3] suggest using MPTCP, but they do not provide practical implementations or performance evaluations, leaving open questions about the real-world feasibility and effectiveness of MPTCP in smart grids environments.

Notably, a QUIC-based multipath transport protocol, Multipath QUIC [35], is currently under development and potentially capable of outperforming MPTCP [36]. However, its progress is less advanced than that of MPTCP, which may explain why, to the best of our knowledge, no research on MPQUIC has been conducted yet in the context of smart grids.

MPTCP Performance. Khalifa et al. [37] present, simulate and evaluate a communication architecture for inter-substation communication using heterogeneous wireless networks with MPTCP, but their study does not cover failovers or real-world hardware. Additionally, numerous performance evaluations of MPTCP have been conducted in other fields [8], [9], [38], [39]. However, only Lopez et al. [9] consider failover delays and their study focusses on a railway application, not accounting for the unique characteristics of large-scale smart grid networks. Furthermore, they do not evaluate different MPTCP implementations (cf. Sec. IV). As a result, existing research highlights MPTCP's potential to enhance resilience in smart grids, but several key questions remain unanswered.

C. Open Research Questions

Overall, previous research has conducted several evaluations of MPTCP and proposed its use for improving resilience in distribution system communication networks. However, there is a lack of conclusive evaluations necessary to determine its feasibility and effectiveness in such networks, particularly with dedicated hardware, in comparison to single-path transport protocols with reliable transmission capabilities, such as TCP and QUIC. Three key research questions arise:

- **RQ1**) Is the deployment of MPTCP on dedicated RTU hardware feasible?
- **RQ2**) How does the performance of MPTCP, TCP and QUIC compare in the event of outages, and what communication delays can we expect in smart grid WANs?
- **RQ3**) Which transport protocol is best suited to fulfill the communication requirements of smart grid applications?

Conducting an evaluation in a real smart grid is impractical due to its critical operation. Furthermore, existing smart grid testbeds [40]–[44] do not include authentic emulations of large-scale WANs. Thus, we develop a testbed based on a real WAN topology which we present in the following section and afterward use to answer the research questions.

Takeaway: *Smart grid WANs need to fulfill critical functionality and redundant communication channels can increase their resilience when combined with effective management. To this end, using MPTCP appears promising, but its feasibility and performance in smart grids remain unclear.*

III. Testbed for Wide Area Communication in Smart Grids

To create our testbed allowing us to investigate the benefits of using MPTCP in smart grids, we used rettij [45], [46], a tool designed for emulating ICT networks and co-simulating smart grid applications. By utilizing containers and Kubernetes for orchestration, rettij allows for scalable emulations, enabling us to replicate the complex real-world topology described below.

A. Emulated Topology

We abstracted an authentic topology —- mirroring the scale and structure —- from the real-world setup of a German DSO through synergies from a joint research project [47]. The WAN topology has a hierarchical structure composed of three layers: backbone, aggregation, and access (cf. Fig. 2). Fiber optics serve as the communication medium, with an assumed latency of 5 µs per km [48].

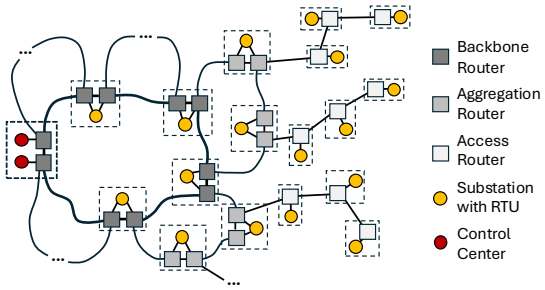


Fig. 2: Hierarchical communication network topology abstracted from the real-world structure of a German DSO.

Backbone Layer. The backbone ring consists of 20 pairs of routers, each with a redundant counterpart, forming a ring structure. The distance between each router pair is approximately 50 km, resulting in a latency of 250 µs. Two control centers (one for redundancy) are connected to one of these backbone router pairs, while the remaining 18 router pairs function as gateways for substations, i.e., 18 substations are directly connected to the backbone ring.

Aggregation Layer. In the aggregation layer, 20 loops consisting of three router pairs are built between two backbone routers, respectively. Therefore, 60 substations are redundantly connected to the aggregation layer. The distance between router pairs in this layer is around 30 km, translating to 150 µs latency.

Access Layer. Last, chains of three individual routers with substations are connected as stubs to aggregation layer routers. The distance between these routers is around 20 km, with a resulting latency of 100 µs.

In total, the topology contains 340 routers and 260 substations, of which 80 are redundantly connected to two gateway routers, forming loops or rings. Each substation operates its own local network, and RTUs within the substations act as gateways for the station IEDs. WAN communication occurs either between the RTUs of different substations or between a substation's RTU and a control center computer.

Based on this topology, we provide a scalable and customizable testbed.

B. Testbed Details

Our testbed includes the following key features.

Scalability. Kubernetes sets a default limit of 110 pods per Kubernetes work machine [49]. However, this limit effectively depends on the underlying hardware and can be adjusted by modifying the `maxPods` and `podCIDR` parameters in the Kubelet configuration [50]. Our testbed, with over 600 nodes and 700 links, successfully operates on a single machine with the hardware specified in Sec. V-A. Horizontal scaling across multiple machines is also possible if needed.

Automated YAML Generation. To reduce implementation complexity and facilitate rapid adjustments, we provide a modular Python script. This script allows for flexible definition

of key parameters, such as the number of nodes in each hierarchical layer and the characteristics of network links. Based on these inputs, the script generates a YAML file, which serves as input for rettij, defining the full topology.

Automated Network Configuration. The testbed configuration script automatically assigns free IPv4 network addresses from a configurable address pool to each generated link. Additionally, a breadth-first search algorithm calculates the optimal next hop (i.e., the fewest overall hops) for each node to reach every destination in the network.

Takeaway: *Our testbed emulates the large-scale topology of a real distribution system's communication network. It is highly customizable, providing an authentic environment for testing smart grid WAN communication concepts and protocols.*

IV. Feasibility Evaluation

To answer RQ1, i.e., assess the deployability of MPTCP, we implemented MPTCP on standard RTU hardware, which is responsible for inter-substation and substation-to-control-center communication.

MPTCP Implementations. Currently, two MPTCP implementations exist: MPTCPv0 [51], which is an experimental out-of-tree version compatible with Linux kernels up to v5.4, and MPTCPv1 [52], which is an upstream version for Linux kernel v5.6 onward, but without support for redundant transmission at this time. We assessed both versions.

RTU Hardware. RTUs typically comprise ARM microprocessors, which support Linux and theoretically enable the use of either MPTCPv0 or MPTCPv1. To confirm that assumption and assess the setup effort, we examined a FW-5-GATE-4G [27] by LACROIX SAE IT-systems (cf. Fig. 1b). The FW-5-GATE-4G is sold and used worldwide, while featuring similar hardware as comparable devices from other manufacturers, e.g., Siemens' SICAM A8000 [53] or Hitachi Energy's RTU product lines [54], making it a representative choice for evaluation. In detail, the FW-5-GATE-4G features an NXP i.MX 6ULL microprocessor with an ARM Cortex-A7 core, 512 MB of RAM, two 100 Mbit Ethernet ports and an integrated LTE modem.

Running MPTCPv0. Since no pre-built MPTCPv0 Linux kernel exists for embedded ARM devices such as the FW-5-Gate-4G, we needed to manually compile a kernel with MPTCP v0.96. To run that kernel, we built a tailored U-Boot bootloader [55], and a root file system using the Embedded Linux Build Environment (ELBE) [56] with Debian 12. However, our kernel did not include some i.MX specific adjustments that are only available in an out-of-tree kernel version maintained by NXP [57], as that would require complex merging of the two out-of-tree kernels. This lack of optimization affected the power management, leading to increased communication latency, but could be resolved for our evaluation by disabling the power management. However, the need and use of optimized i.MX kernels should be assessed before real-world deployment. Furthermore, the lack of security patches in outdated out-of-tree kernel versions should be taken into account.

Running MPTCPv1. MPTCPv1 is officially supported by Linux kernels since v5.6. As such, it is easily configurable when building a respective kernel, including optimized i.MX kernels. We tested this assumption for the FW-5-Gate-4G and successfully installed a v6.1 kernel with MPTCPv1. However, older devices and the need for redundant scheduling still necessitate the use of MPTCPv0.

Finally, both MPTCP versions ran smoothly on our FW-5-Gate-4G and operated in an application-transparent manner, automatically managing multiple interfaces without requiring manual intervention compared to TCP and QUIC, as demonstrated in our subsequent evaluation.

Takeaway: *Deploying MPTCP on embedded RTU hardware requires moderate effort, but our evaluation indicates that it is compatible with a broad range of devices.*

V. Failover Performance of Transport Protocols in Smart Grids

To address RQ2, i.e., assess the failover delays of MPTCP, TCP, and QUIC, we connected the RTU (cf. Sec. IV) and a physical machine acting as the control center computer (CCC) to our testbed. We then added a wireless backup channel based on LTE due to its prevalence in today's hardware and deployments. Given the high variability of wireless connections (cf. Sec. V-B1), we additionally implemented an Ethernet-based backup channel to gain deeper insights. We transmitted packets between CCC and RTU, measuring their Round-Trip Time (RTT) while simulating link failures. Below, we first describe our evaluation setup, followed by a discussion of the results.

A. Evaluation Setup

Fig. 3 presents an overview of our setup.

Hardware. We deployed our testbed on a single machine equipped with two AMD EPYC 7443 24-core processors, 512 GB of RAM, and NVMe SSD storage. The machine ran Ubuntu 22.04.3 LTS with Linux kernel v5.15 and provided two 25 Gbit Ethernet interfaces. The CCC ran Ubuntu 22.04.4 LTS and was equipped with an Intel Core i7-12700 processor, 32 GB of RAM, solid-state NVMe storage and a Gbit interface. Additionally, we connected a Huawei E3372h-320 LTE dongle to the CCC via USB, enabling a cellular backup connection.

Network Setup. We connected the testbed server, RTU and CCC to our institute's network, isolating their traffic with VXLAN-based overlay networks on layer 2. We then bridged the VXLAN interfaces of RTU and CCC to the network emulation, which also used VXLANs due to rettij's design [46]. To evaluate the worst-case latency evaluation, we connected the CCC to one of the backbone routers (cf. Sec. III-A) and the RTU to one of the chained routers in the access layer on

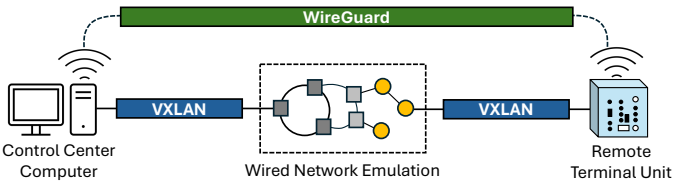


Fig. 3: Hardware and network setup of our evaluation.

the opposite side of the backbone ring, creating a path with 23 hops — the maximum possible distance. Additionally, we connected the RTU and CCC through a backup channel.

LTE Backup Channel. Due to the absence of a private cellular network, we utilized public LTE infrastructure. However, internal firewalls block inbound connections and connections from within the mobile network operators' network. To establish a connection between two LTE clients nevertheless, we used SIM cards from two different operators and performed hole punching. To facilitate the hole punching and separate it from the evaluation traffic, we employed WireGuard [58], which tunnels Layer 3 traffic between both devices and implicitly performs hole punching by regularly triggering them to reach each other. While WireGuard adds an constant latency overhead of around 0.6 ms, it leaves all results comparable.

Ethernet Backup Channel. To gain further insights, we also established a backup channel using WireGuard and the Ethernet connection between CCC and RTU, bypassing the emulated network.

Measurement Scripts. We developed Python-based client and server applications to perform the measurements. These were deployed on both the RTU and CCC. We utilized the *asyncio* module for TCP and MPTCP and the *aioquic* module for QUIC. In line with future communication requirements [4], the client continuously sent packets with a 1000 B payload every 0.05 s to the server, which echoed the packets back to the client, enabling the client to measures their RTTs. For TCP and QUIC, we implemented a timeout to realize failovers. This triggers connection re-establishment over the backup channel and retransmission of unacknowledged packets.

Protocol Configurations. To minimize delays, we disabled Nagle's algorithm for TCP. For MPTCP, we set the path manager to `fullmesh` and the number of subflows (`num_subflows`) to 1, which creates exactly one subflow for every pair of IP addresses, ensuring that one subflow used Ethernet and another the backup connection.

Encryption. Since QUIC includes built-in TLS 1.3 encryption, introducing additional latency for key exchange and encryption, we added TLS 1.3 to both TCP and MPTCP for a fair comparison. We used 2048 bit RSA certificates for all protocols and evaluated TCP and MPTCP with and without encryption.

Link Failures. To simulate connection outages, our client script triggered the dropping of all packets in the firewall after a random interval between 15 s and 30 s. Hence, the networking stack remained unaware of the failure, ensuring the results were not distorted. After another random interval of 15 s to 30 s, the wired connection was restored. Additional implementation details can be found in our project repository [12].

B. Measurements

In real-world scenarios, the RTU typically acts as the client and the CCC as the server. However, due to the RTU's limited processing and storage capacity, we reversed these roles in our setup. We conducted 30 runs of 60 seconds each for every evaluation, calculating the average, median, and the 1st and 99th percentiles across all runs. We began with a baseline measurement using plain TCP and then evaluated the RTT for the transport protocols and configurations listed in Tab. I, including both backup channels. We did not observe significant differences caused by using other congestion control algorithms, thus sparing their presentation.

TABLE I: Evaluated transport protocols configurations including Congestion Control (CC), Security and Timeouts.

#	Protocol	CC-Alg.	Security	Timeout [ms]
1	TCP	Cubic	-	50
2	TCP	Cubic	TLS 1.3	50
3	QUICv1	Cubic	TLS 1.3	50
4	MPTCPv0 (def. sched.)	Balia	-	-
5	MPTCPv0 (def. sched.)	Balia	TLS 1.3	-
6	MPTCPv1 (def. sched.)	Cubic	-	-
7	MPTCPv1 (def. sched.)	Cubic	TLS 1.3	-
8	MPTCPv0 (red. sched.)	Balia	-	-
9	MPTCPv0 (red. sched.)	Balia	TLS 1.3	-

1) Baseline Measurements: The baseline results are shown in Tab. II and Fig. 4, representing the best case RTT for each evaluated channel and revealing the potential performance limitations of the protocols and configurations in subsequent tests. The LTE-based backup channel exhibited fluctuating RTT above 80 ms, which was significantly higher than the RTT over the emulated network. In contrast, the RTTs of the Ethernet-based backup channel had much less deviation, averaging 5 ms lower than the RTT over the emulated network.

2) Basic Evaluation with Ethernet-based Backup: We first examined failover times using the stable Ethernet-based backup connection, assuming that the emulated WAN network was the primary communication link between the CCC and RTU. This connection was taken down at a random point in time during testing. Notably, the backup connection had lower latency than the main connection in this scenario (as shown in the baseline measurements).

Fig. 5 illustrates the aggregated RTTs for packets successfully transmitted before the failover, the RTT for the first packet impacted by the failover (referred to as the *failover time*), and

TABLE II: Baseline measurements of the RTT for the emulated WAN and backup connections via LTE and Ethernet [ms].

Connection	Avg.	Mdn.	0.01 quant.	0.99 quant.
Emul. WAN	8.018	8.184	5.064	11.053
Backup LTE	86.601	81.067	60.324	157.903
Backup Eth.	3.602	3.576	2.589	5.27

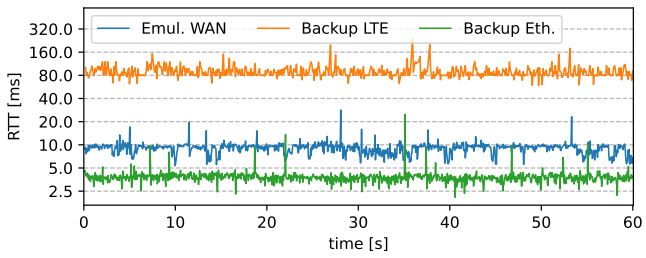


Fig. 4: Sample baseline measurements of the different paths between CCC and RTU, i.e., emulated WAN network, public LTE infrastructure and direct Ethernet connection

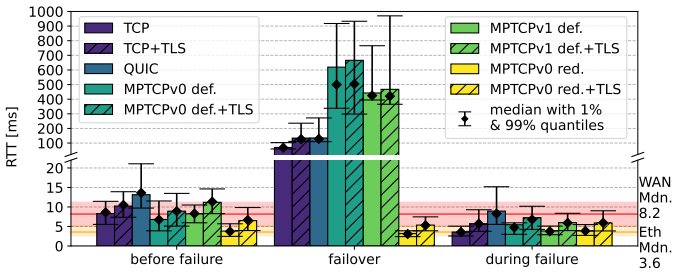


Fig. 5: Measured RTTs for the failover, as well as before and during the induced failure of the emulated WAN, when using an Ethernet-based backup connection and a 50 ms timeout for both TCP and QUIC.

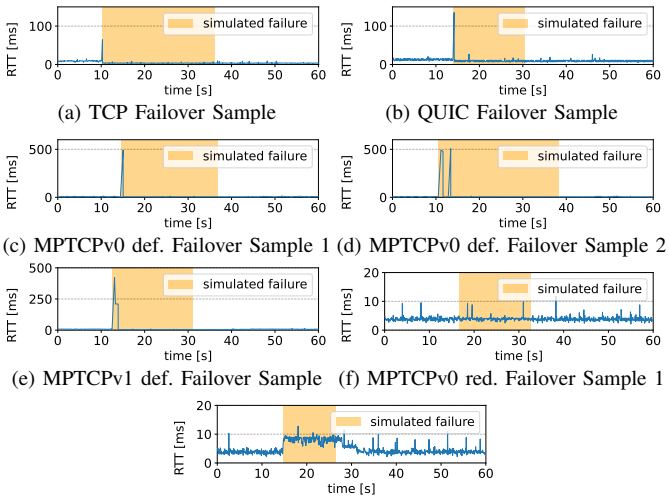
the aggregated RTTs for the subsequent packets transmitted while the outage endured.

TCP and QUIC. Before the link failure, plain TCP's RTT matched the baseline while using TLS increased the RTT by around 2 ms. QUIC added an additional 3 ms to the RTT, which we attribute to its less efficient Python-based user space implementation.

TCP and QUIC only switched to the backup connection after a manually configured timeout was exceeded. A timeout of 20 ms (around 2x the RTT of the main connection TCP+TLS) was still too short for our TLS scenario, causing premature switching before an actual failure. Therefore, to ensure comparability, we increased the timeout to 50 ms, resulting in a single, visible failover switch at the outage start (cf. Fig. 6a and Fig. 6b). However, increasing the timeout simply leads to a corresponding increase in RTTs at failure, which helped us evaluate the effect of different timeouts, discussed in Sec. V-B4.

Theoretically, TCP's failover time consists of the configurable timeout plus 1.5xRTT, where the RTT of the backup channel includes establishing a new connection (without TLS) and retransmitting the first lost packet. TLS adds additional latency due to encryption negotiations [59]. QUIC, which implements encryption by default, re-establishes connections faster due to its 0-RTT resumption [60]. However, in our test, both TCP+TLS and QUIC had an average RTT of around 134 ms, with medians of around 127 ms. This was again likely due to QUIC's user-space implementation. Plain TCP showed an averaged RTT of 70 ms with a median of 68 ms. After switching to the backup connection, RTTs for unencrypted connections returned to baseline levels, with a slight increase for TCP+TLS and QUIC.

MPTCPv0 and MPTCPv1 - Default Scheduler. MPTCP's default scheduler establishes two subflows, one over each interface, but uses only one subflow actively at a time. Interestingly, in our test case, the default scheduler of MPTCPv0 switched multiple times between both subflows *before* the failure occurred, resulting in average RTTs lying between the baselines of the two connections (cf. Fig. 6c). This behavior is likely due to both connections having similar latencies, making it difficult for the scheduler to consistently prefer one subflow over the other as the default scheduler constantly evaluates all available subflows to use the one with the lowest RTT [61].



(a) TCP Failover Sample (b) QUIC Failover Sample

(c) MPTCPv0 def. Failover Sample 1 (d) MPTCPv0 def. Failover Sample 2

(e) MPTCPv1 def. Failover Sample (f) MPTCPv0 red. Failover Sample 1

(g) MPTCPv0 red. Failover Sample 2

Fig. 6: RTTs before, during, and after emulating an outage of the main connection (marked in orange).

Furthermore, the default scheduler sometimes switched back to the failed subflow briefly, causing another delay of a similar magnitude (cf. Fig. 6d). Here, we assume that this first switch was random, and the scheduler only marked the original path as failed afterward.

Strikingly, MPTCPv1 did not expose this behavior (cf. Fig. 6e) and had significantly lower failover RTTs of 442 ms on average with a median of 424 ms without TLS. With TLS, the average RTT was about 467 ms with a median of 420 ms. In contrast, the failover process typically took 619 ms on average, with a median of 499 ms, for MPTCPv0 without TLS, and 665 ms on average, with a median of 504 ms, with TLS. However, with both versions, the RTT was not only increased for the first packet affected by the link failure but also for up five subsequent packets (cf. Fig. 6c, Fig. 6d, and Fig. 6e).

Still, our evaluation shows that both MPTCP versions enable automatic failover independent of the application. Additionally, they detect when the previously failed subflow becomes available again, eventually switching back to it.

MPTCPv0 - Redundant Scheduler. With MPTCPv0's redundant scheduler, all packets were delivered over the designated backup connection even before the failure occurred (cf. Fig. 6f), as it had lower latency than the main connection in this scenario. While this may seem counterintuitive, it highlights a likely scenario when using ultra-reliable, low-latency media like 5G or 6G, expected to be deployed in future industrial networks.

This example also underscores the redundant scheduler's main drawback, which is communication overhead as the same data occupies two channels. On the other hand, all messages arrived reliably with the lowest possible latency, without any interference, when the WAN connection failed.

To evaluate the redundant scheduler's operation in case the best subflow fails, we also emulated an outage of the Ethernet connection while the WAN operated correctly. In this case, no significant failover delay occurred, and the RTT simply increased to reflect the inherent latency of the backup medium

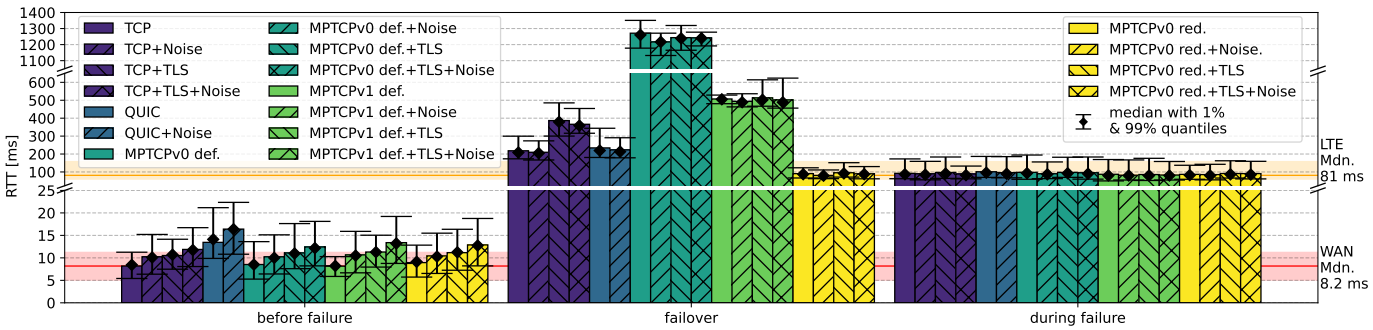


Fig. 7: Measured RTTs for the failover, as well as before and during the induced failure of the emulated WAN, when using an LTE-based backup connection and a 50 ms timeout for both TCP and QUIC.

(cf. Fig. 6g), as the redundant scheduler was already using both subflows actively. Interestingly, the primary subflow was not immediately restored when the failed path became available again, but only after a delay of around 5 seconds.

3) Real-World Evaluation: In recent years, private LTE networks have been deployed globally for mission-critical communication, including smart grid applications [62], [63]. Thus, to assess performance in a real-world scenario, we conducted an evaluation using the RTU's integrated LTE module and the LTE-based backup channel described in Sec. V-A alongside background traffic from other substations over the emulated WAN.

Background Traffic. Assuming a high-performance application scenario such as wide area protection [4], with 15 IEDs per substation sending Manufacturing Message Specification (MMS) messages of approximately 157 B every 0.1 s, we expect at least 40k messages per second reaching the CCC over the WAN. We hence implemented a generous number of 50k messages that were received by the CCC every second.

General Observations. The background traffic generally increased RTTs through the WAN by up to 3 ms (cf. Fig. 7), letting them rise to over 10 ms with plain TCP and 12 ms and 13.5 ms for TCP+TLS and QUIC, respectively, before the failure when all substations continuously exchanged data with the CCC. As observed in the baseline measurement, LTE exhibited significantly higher RTTs (around 80 ms) and greater variance in failover times. Overall, most observations from the basic evaluation were consistent, with two notable exceptions.

TCP and QUIC. In this scenario, QUIC's failover time was significantly lower than TCP+TLS, unlike in previous tests. We attribute this to QUIC's 0-RTT resumption feature, which provided a significant advantage given LTE's higher base RTT. On average, QUIC showed a failover RTT of 234 ms, with a median of 219 ms, similar to plain TCP (218 ms average, 208 ms median). In contrast, TCP+TLS had an average failover RTT of 386 ms, with a median of 379 ms.

MPTCPv0 and MPTCPv1. MPTCPv0's redundant scheduler continued to offer seamless failover, with an average RTT of around 95 ms and a median of 91 ms. Unlike with the Ethernet-based backup connection, the default scheduler of MPTCPv0 no longer switched between subflows, likely due to the now-clear performance difference between the two connections. However, it exhibited an even higher failover delay — over 1.2 s with or without TLS — providing similar values for average and median. In contrast, MPTCPv1's default scheduler exposed only slightly higher failover delays, with average and median RTTs around 500 ms.

4) Significance of Results: Our evaluation aligns with findings from Lopez et al. [9], while offering additional insights into the behavior of different transport protocols, including both MPTCP versions, during network failures in smart grid WANs. We specifically examined MPTCP's performance over two types of backup connections with varying latencies: one with latency slightly lower than the main WAN and another with significantly higher latency. To this end, our evaluation confirms that MPTCP's redundant scheduler reliably selects the lowest-latency path, ensuring best-case performance at all times. Lopez et al. already demonstrated that the default MPTCP scheduler incurs a significant delay before utilizing the backup subflow when the active one fails. Our results further show that the default scheduler of MPTCPv0 may even switch back to a failed subflow if both subflows have similar characteristics, causing additional delays due to a second failover, while the default scheduler of MPTCPv1 provides significantly lower failover delays, especially when using LTE as backup medium. Last, we show that MPTCP automatically switches back to the previously failed path once it becomes available again, though with a slight delay of several milliseconds.

Regarding TCP and QUIC, we observed a trade-off between both MPTCP schedulers. QUIC, even with its Python-based implementation, performed better in the high-latency LTE environment, likely due to its 0-RTT resumption. With implementations closer to the hardware (e.g., in C/C++), we expect even greater performance improvements. However, the failover time is strongly influenced by the selected timeout value (set to 50 ms in our evaluation). In the presence of background traffic, QUIC's RTT was around 13 ms, and TCP+TLS's RTT was around 10 ms in the emulated WAN. Thus, a lower timeout value of 2x RTT, such as 26 ms for QUIC and 20 ms for TCP+TLS, instead of 50 ms reduces their failover time, respectively. However, as our evaluation also shows, lower timeouts risk premature switching without actual link failures, necessitating thorough testing to determine the optimal value.

In addition to protocol-specific observations, our evaluation showed that background traffic — present in real networks —

can generally increase latencies, which must be factored in for time-critical use cases such as tele-protection (cf. Sec. II-A). Thus, overall, the failover performance of the protocols is highly dependent on the specific scenario, communication medium, and configuration.

Takeaway: *MPTCP's redundant scheduler consistently provides the lowest possible latencies, while MPTCP's default scheduler performs worse than both TCP and QUIC.*

VI. Suitability Analysis and Future Work

Subsequently, we answer RQ3, i.e., compare the suitability of MPTCP, TCP and QUIC for providing resilient communication in smart grids, by discussing the implications of our findings.

Based on our practical evaluations, we conclude that MPTCPv1 with the default scheduler, supported by Linux kernels since v5.6, offers the most convenient solution in terms of effort and overhead. However, it is only viable if RTTs of over 400 ms are temporarily acceptable. While TCP and QUIC allow for lower latencies, their use for failovers requires vendor implementation in proprietary applications and sensible configuration of timeouts to avoid unnecessary switches to less performant channels. Although mechanisms like load balancing and DNS can facilitate transparent interface switching, the application must still re-establish the connection and retransmit lost packets. A similar process is necessary to reuse previously failed paths, rendering broad adoption of TCP and QUIC with backup channels cumbersome without widespread vendor support and implementation. Additionally, our calculations show that meeting one-way latency requirements of less than 50 ms becomes unfeasible in the event of link failures when applying TLS, even with a short timeout of 20 ms and backup channels with only 3 ms RTT. Given the increasing frequency of cyber-attacks, encryption and authentication are essential for smart grids [64], especially when wireless communication channels are involved [65].

Consequently, redundant transmission, as implemented by MPTCPv0's redundant scheduler, is the only option to meet latency requirements below 10 ms as required by tele-protection (cf. Sec. II-A), provided it is paired with a suitable backup channel. Previous measurements in other industrial contexts [66] suggest that RTTs as low as 35 ms are achievable with private LTE networks, though they also show a significant variability, which indicated that consistently meeting one-way latency requirements of 20 ms with LTE is rather unlikely. However, recent research on private industrial 5G networks [67] suggests RTTs ranging from 12 ms to 40 ms, with further improvements likely in the future. As a result, a combination of fiber optics and 5G, along with MPTCPv0's redundant scheduling, presents a promising solution for ensuring both resilience and low latency, already meeting 20 ms one-way requirements reliably, and often achieving below 10 ms. Still, elaborate tests of individual setups are required to obtain concrete values in the respective scenario. In the future, 6G networks may provide even lower latencies, potentially in the range of milliseconds, thus reliably meeting the most stringent latency requirements.

A. Limitations and Future Work

Although our feasibility evaluation indicates that MPTCPv0 with redundant scheduling is broadly compatible with currently deployed RTU devices, significant effort and expert knowledge are required to build a kernel with i.MX specific optimizations. Therefore, implementing the redundant scheduler in MPTCPv1 could significantly increase its accessibility and simplify deployment, particularly given its benefits for time-critical communication with high availability requirements. However, using two network interfaces and channels redundantly introduces overhead in the backup network, which might even exceed its capacity if redundant scheduling is applied to every RTU. Thus, further research is necessary to confirm the compatibility of private LTE/5G networks with large-scale inter-substation and substation-to-control center communication. Additionally, the development of a variant of the default MPTCP scheduler, which reacts more quickly to the failure of an active subflow, could offer a more balanced trade-off between efficiency and performance. Such a solution could provide similar failover latencies to TCP and QUIC but without their associated implementation overhead. Given the critical importance of network security and the greater vulnerability of wireless backup channels compared to underground fiber optic cables, further research is needed to assess the viability of these channels for transmitting critical data on a large scale and to develop robust security measures.

Takeaway: *MPTCP with redundant scheduling is the only option to meet latency requirements in the range of a few milliseconds in the case of large-scale network failures. If temporal RTTs over 400 ms are acceptable, the default scheduler provides the best option due to its availability in modern Linux Kernels and lower overhead. Kernel implementation of the redundant scheduler and adaptations of the default scheduler might mitigate their respective disadvantages in the future.*

VII. Conclusion

This paper investigates the capabilities of single- and multi-path transport protocols for ensuring resilient WAN communication in future smart grids while meeting their stringent latency requirements. We evaluated the performance and suitability of TCP, QUIC, and MPTCP through an emulation of an authentic distribution grid WAN, conducting comprehensive tests on the RTT between a physical CCC and a real RTU while simulating connection outages. Our results provide transparency to DSOs, enabling them to make an informed decision matching their needs and ensure seamless operation of their grid. While MPTCP's redundant scheduler operates transparently to applications and proves to be the most reliable option when millisecond-level latencies are essential at all times, future optimizations of the default scheduler could provide a better balance between efficiency and performance.

Acknowledgments

Funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) — Research Project VeN2uS — 03EI6053K.

References

[1] X. Lu, W. Wang, and J. Ma, "An empirical study of communication infrastructures towards the smart grid: Design, implementation, and evaluation," *IEEE Trans. Smart Grid*, vol. 4, pp. 170–183, Mar. 2013.

[2] N. Dorsch, H. Georg, and C. Wietfeld, "Analysing the real-time-capability of wide area communication in smart grids," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 682–687, IEEE, Apr. 2014.

[3] 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.

[4] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Communication network requirements for major smart grid applications in HAN, NAN and WAN," *Computer Networks*, vol. 67, pp. 74–88, July 2014.

[5] M. G. Adamiak, A. P. Apostolov, M. M. Begovic, C. F. Henville, K. E. Martin, G. L. Michel, A. G. Phadke, and J. S. Thorp, "Wide area Protection—Technology and infrastructures," *IEEE Trans. Power Delivery*, vol. 21, pp. 601–609, Apr. 2006.

[6] M. Shahraeini and P. Kotzanikolaou, "Resilience in wide area monitoring systems for smart grids," in *Wide Area Power Systems Stability, Protection, and Security* (H. Haes Alhelou, A. Y. Abdelaziz, and P. Siano, eds.), pp. 555–569, Cham: Springer International Publishing, 2021.

[7] Q. Yang, J. A. Barria, and T. C. Green, "Communication infrastructures for distributed control of power distribution networks," *IEEE Trans. Ind. Inf.*, vol. 7, pp. 316–327, May 2011.

[8] C. H. Park, P. Austria, Y. Kim, and J.-Y. Jo, "MPTCP performance simulation in multiple LEO satellite environment," in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0895–0899, IEEE, Jan. 2022.

[9] I. Lopez, M. Aguado, C. Pinedo, and E. Jacob, "SCADA systems in the railway domain: Enhancing reliability through redundant MultipathTCP," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 2305–2310, IEEE, Sept. 2015.

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

[11] "Redudant scheduler for the MPTCP out-of-tree implementation." https://github.com/multipath-tcp/mptcp/blob/mptcp_v0.96/net/mptcp/mptcp_redundant.c.

[12] I. Fink, L. Ferlemann, M. Dahlmanns, C. Thimm, and K. Wehrle, "Implementation of our topology with rettij." https://github.com/COMSYS/distribution-grid-emulation, 2025.

[13] S. Mohagheghi, J. Stoupis, and Z. Wang, "Communication protocols and networks for power systems-current status and future trends," in *2009 IEEE/PES Power Systems Conference and Exposition*, pp. 1–9, IEEE, Mar. 2009.

[14] L. Bader, M. Serror, O. Lamberts, Ö. Sen, D. van der Velde, I. Hacker, J. Filter, E. Padilla, and M. Henze, "Comprehensively analyzing the impact of cyberattacks on power grids," in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pp. 1065–1081, IEEE, July 2023.

[15] 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.

[16] Dominique Verhulst, "IP/MPLS secures the journey to a net-zero emissions future." https://www.smart-energy.com/industry-sectors/smart-grid/ip-mpls-secures-the-journey-to-a-net-zero-emissions-future/, October 2022.

[17] Cisco Systems, Inc., "Why IP Is the Right Foundation for the Smart Grid." https://www.cisco.com/c/dam/en_us/solutions/industries/docs/energy/c11-581079-wp.pdf, 2010.

[18] K.-H. Mak and B. L. Holland, "Migrating electrical power network SCADA systems to TCP/IP and ethernet networking," *Power Eng. J.*, vol. 16, pp. 305–311, Dec. 2002.

[19] Z. H. E. Hossain and H. P. (eds.), *Smart Grid Communications and Networking*. Cambridge University Press, 2012.

[20] M. A. Ridwan, N. A. M. Radzi, W. S. H. M. Wan Ahmad, F. Abdullah, M. Z. Jamaludin, and M. N. Zakaria, "Recent trends in MPLS networks: technologies, applications and challenges," *IET Commun.*, vol. 14, pp. 177–185, Jan. 2020.

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

[22] K. Shahid, A. Saeed, T. le Fevre Kristensen, and R. L. Olsen, "Impact of transport layer protocols on reliable information access in smart grids," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp. 1–6, IEEE, Sept. 2017.

[23] N. S. Nafi, K. Ahmed, M. A. Gregory, and M. Datta, "A survey of smart grid architectures, applications, benefits and standardization," *J. Netw. Comput. Appl.*, vol. 76, pp. 23–36, Dec. 2016.

[24] 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.

[25] K. Ghanem, S. Ugwuanyi, R. Asif, and J. Irvine, "Challenges and promises of 5g for smart grid teleprotection applications," in *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–7, 2021.

[26] R. Bächli, M. Häusler, and M. Kranich, "Teleprotection solutions with guaranteed performance using packet switched wide area communication networks," in *2017 70th Annual Conference for Protective Relay Engineers (CPRE)*, pp. 1–6, 2017.

[27] "FW-5-GATE-4G." https://www.sae-it.com/product/net-line-fw-5-gate-4g/.

[28] J. P. Astudillo León and L. J. de la Cruz Llopis, "A joint Multi-Path and Multi-Channel protocol for traffic routing in smart grid neighborhood area networks," *Sensors*, vol. 18, Nov. 2018.

[29] I. Ali and M. S. Thomas, "Substation communication networks architecture," in *2008 Joint International Conference on Power System Technology and IEEE Power India Conference*, pp. 1–8, IEEE, Oct. 2008.

[30] T. Duan and V. Dinavahi, "Fast path recovery for single link failure in SDN-Enabled wide area measurement system," *IEEE Trans. Smart Grid*, vol. 13, pp. 1645–1653, Mar. 2022.

[31] M. Elattar, V. Wendt, A. Neumann, and J. Jasperneite, "Potential of multipath communications to improve communications reliability for internet-based cyberphysical systems," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8, IEEE, Sept. 2016.

[32] Y. Hong, D. Kim, D. Li, L. Guo, J. Son, and A. O. Tokuta, "Two new multi-path routing algorithms for fault-tolerant communications in smart grid," *Ad Hoc Networks*, vol. 22, pp. 3–12, Nov. 2014.

[33] K. M. Muttaqi, J. Aghaei, V. Ganapathy, and A. E. Nezhad, "Technical challenges for electric power industries with implementation of distribution system automation in smart grids," *Renewable Sustainable Energy Rev.*, vol. 46, pp. 129–142, June 2015.

[34] S. Zhang and V. Vittal, "Wide-Area control resiliency using redundant communication paths," *IEEE Trans. Power Syst.*, vol. 29, pp. 2189–2199, Sept. 2014.

[35] Y. Liu, Y. Ma, Q. D. Coninck, O. Bonaventure, C. Huitema, and M. Kühlewind, "Multipath Extension for QUIC," Internet-Draft draft-ietf-quic-multipath-11, Internet Engineering Task Force, Oct. 2024. Work in Progress.

[36] Q. De Coninck and O. Bonaventure, "Multipath QUIC: Design and evaluation," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, (New York, NY, USA), ACM, Nov. 2017.

[37] T. Khalifa, A. Abdrabou, K. Shaban, and A. M. Gaouda, "Heterogeneous wireless networks for smart grid distribution systems: Advantages and limitations," *Sensors*, vol. 18, May 2018.

[38] M. Elattar, M. Friesen, and J. Jasperneite, "Evaluation of multipath communication protocols for reliable internet-based cyber-physical systems," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1195–1200, IEEE, June 2017.

[39] E. Dong, M. Xu, X. Fu, and Y. Cao, "A loss aware MPTCP scheduler for highly lossy networks," *Computer Networks*, vol. 157, pp. 146–158, July 2019.

[40] D. van der Velde, Ö. Sen, and I. Hacker, "Towards a scalable and flexible smart grid Co-Simulation environment to investigate communication infrastructures for resilient distribution grid operation," in *2021 Interna-*

tional Conference on Smart Energy Systems and Technologies (SEST), pp. 1–6, IEEE, Sept. 2021.
[41] C. Hannon, J. Yan, D. Jin, C. Chen, and J. Wang, "Combining simulation and emulation systems for smart grid planning and evaluation," *ACM Trans. Model. Comput. Simul.*, vol. 28, pp. 1–23, Aug. 2018.
[42] H. Palahalli, E. Ragaini, and G. Gruosso, "Smart grid simulation including communication network: A hardware in the loop approach," *IEEE Access*, vol. 7, pp. 90171–90179, 2019.
[43] S. Tan, W. Song, D. Huang, Q. Dong, and L. Tong, "Distributed software emulator for Cyber-Physical analysis in smart grid," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 506–517, 2017.
[44] C. Holt, A. Kong, A. St. Leger, and D. Bennett, "Communications network emulation for smart grid test-bed," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pp. 1–5, IEEE, July 2016.
[45] F. Niehaus, B. Fraune, G. Gritzan, and R. Sethmann, "Modern ICT network simulator for co-simulations in smart grid applications," *iccws*, vol. 17, pp. 227–236, Mar. 2022.
[46] "rettij network simulator." https://gitlab.com/frihsb/rettij, 2021.
[47] "Vernetzte Netzschutzsysteme (VeN2uS) Projektwebseite." https://ven2us.de.
[48] CommScope, "Latency in optical fiber systems." https://www.commscope.com/globalassets/digizuite/2799-latency-in-optical-fiber-systems-wp-111432-en.pdf.
[49] "Kubernetes scalability thresholds." https://github.com/kubernetes/community/blob/master/sig-scalability/configs-and-limits/thresholds.md.
[50] "Kubelet configuration (v1beta1)." https://kubernetes.io/docs/reference/config-api/kubelet-config.v1beta1/, 2024.
[51] Paasch, C., Barre, S., et al., "Multipath TCP in the Linux Kernel, available from https://www.multipath-tcp.org." https://www.multipath-tcp.org.
[52] "Multipath TCP for Linux." https://www.mptcp.dev.
[53] "Siemens SICAM A8000." https://emaselsewedy.com/wp-content/uploads/2020/04/Smart-Solution-Catalog.pdf, 2024.
[54] "Hitachi Energy RTU Product Lines." https://www.hitachienergy.com/products-and-solutions/substation-automation-protection-and-control/products/remote-terminal-units.
[55] "U-boot." https://source.denx.de/u-boot/u-boot.
[56] "Embedded linux build environment." https://elbe-rfs.org.
[57] "i.MX Software and Development Tools." https://www.nxp.com/design/design-center/software/embedded-software/i-mx-software:IMX-SW.
[58] "Wireguard." https://www.wireguard.com, 2022.
[59] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3." RFC 8684, 2018.
[60] M. Thomson and S. Turner, "Using TLS to Secure QUIC." RFC 9001, May 2021.
[61] "Configure MPTCP." https://multipath-tcp.org/pmwiki.php/Users/ConfigureMPTCP.
[62] "The Power of LTE 450 for Critical Infrastructure." https://450alliance.org/the-power-of-lte-450-for-critical-infrastructure/.
[63] "Utilities shift from tests to deployments with private LTE." https://www.lightreading.com/security/utilities-shift-from-tests-to-deployments-with-private-lte.
[64] Z. E. Mrabet, N. Kaabouch, H. E. Ghazi, and H. E. Ghazi, "Cyber-security in smart grid: Survey and challenges," *Computers & Electrical Engineering*, vol. 67, pp. 469–482, 2018.
[65] M. Wen, Q. Li, K. J. Kim, D. López-Pérez, O. A. Dobre, H. V. Poor, P. Popovski, and T. A. Tsiftsis, "Private 5G Networks: Concepts, Architectures, and Research Landscape," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 1, pp. 7–25, 2022.
[66] E. Lyczkowski, H. A. Munz, W. Kiess, and P. Joshi, "Performance of private lte on the factory floor," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2020.
[67] T. Lackner, J. Hermann, F. Dietrich, C. Kuhn, M. Angos, J. L. Jooste, and D. Palm, "Measurement and comparison of data rate and time delay of end-devices in licensed sub-6 ghz 5g standalone non-public networks," *Procedia CIRP*, vol. 107, pp. 1132–1137, 2022. Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022.