Published in: GLOBECOM 2016 - 2016 IEEE Global Communications Conference
DOI: 10.1109/GLOCOM.2016.7841813

Enhanced Fast Failover for Software-Defined Smart Grid Communication Networks

Nils Dorsch, Fabian Kurtz, Felix Girke, Christian Wietfeld
Communication Networks Institute, TU Dortmund, Otto-Hahn-Strasse 6, 44227 Dortmund
Email: {nils.dorsch, fabian.kurtz, felix.girke, christian.wietfeld}@tu-dortmund.de

***Abstract*—Future energy systems depend on a corresponding Information and Communication Technology (ICT) overlay, allowing timely transmission of critical grid information in particular in case of failures or other unanticipated events. Therefore, to maintain grid stability, Smart Grid communication networks are required to be highly reliable and real-time capable. Within this paper, we propose and evaluate techniques for improved fault tolerance with regard to link failures within the ICT infrastructure, utilizing the concepts of Software-Defined Networking (SDN). Centralized and decentralized approaches for both failure detection and traffic recovery are compared. While decentralized approaches, employing Bidirectional Forwarding Detection and OpenFlow's fast failover groups, allow for shorter traffic disruptions by saving the delay of controller-switch communication, they result in sub-optimal configurations and possibly overloaded links. Vice versa, controller-driven approaches, using a custom heartbeat detection mechanism, may offer better alternative configurations due to fast re-calculation of routes, yet incur higher delays. Combining the advantages of both approaches, a hybrid concept is proposed, which enables nearly instant local recovery, succeeded by immediate optimal route updates, issued by the SDN controller. Thus, failover times are reduced to 4.5 ms mean delay, fulfilling IEC 61850 Smart Grid requirements, while providing optimal routes almost continuously and maintaining defined Quality-of-Service (QoS) levels.**

I. Introduction

Sparked by the change towards sustainable energy generation, current energy systems experience fundamental transformations to Smart Grids. A crucial factor in creating Smart Grids is the provision of a suitable ICT infrastructure, matching the specific needs of this environment [1], as shown in Figure 1. First, this infrastructure needs to deal with a variety of requirements induced by use cases like Automated Meter Reading, Eletric Vehicle charging or monitoring and controlling power flows across all voltage levels of the grid. Different use cases bring along a diversity of protocols and access technologies. Therefore, service prioritization is essential for communication in Smart Grids, even more so if public infrastructures are considered for shared use. Especially, monitoring and protection on transmission grid level call for real-time capable ICT infrastructures, enabling the transfer of time-critical measurement data and switching commands. Furthermore, these communication networks need to be extremely fault-tolerant, guaranteeing fast recovery in case of failure. Here, SDN offers a (cost-)effective, yet reliable alternative to the conventional approach of resource over-provisioning. In this paper we focus on the aspect of fault-tolerance mechanisms on basis of SDN. By separating control and data plane, SDN introduces a new level of flexibility to communication networks [2]. A programmable controller allows for dynamic configuration of network devices. In contrast to traffic engineering approaches such as Multipath Label Switching (MPLS), SDN profits from its openness and extensibility, enabling the combination of various concepts. Though OpenFlow (OF) [3] is not the only protocol for controller-switch communication, it is a de-facto standard for SDN.

By comparing central and local approaches of failure detection and recovery, building on SDN and OpenFlow, we follow up the work of [4]. In particular, we provide a hybrid method for fast failure detection and include mechanisms for traffic optimization, subsequent to failures. Our concepts are studied by means of two crucial Smart Grid scenarios: 1) substation automation 2) Wide Area Monitoring, Protection And Control (WAMPAC). Therefore, reference topologies and IEC 61850 [5] standard compliant communication services are applied. IEC 61850 is a standard originally specified for substation automation, which has been extended subsequently to cover most aspects of Smart Grid communication.

This paper is structured as follows: Section III outlines our concepts of fast failure detection and recovery, contrasted with related work in Section II. The test environment used for evaluating the proposed algorithms is described in Section IV. While Section V introduces the application scenarios, Section VI provides selected analysis results. Finally, Section VII presents a conclusion and an outlook on future work.

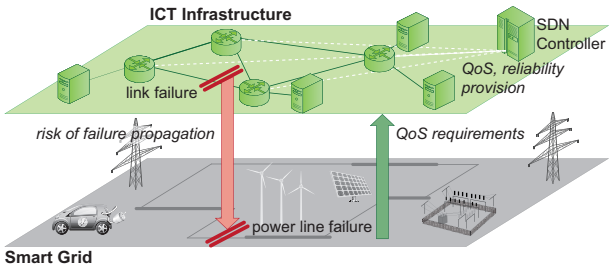

Figure 1: Failures in Integrated ICT-Smart Grid-Infrastructures with SDN for Fast Recovery and QoS Provision

II. Related Work

In recent years, SDN has become a promising and controversially discussed approach for future communication networks. However, only few works drew the connection to the

issue of upcoming Smart Grid communications. Sydney et al. proposed the application of SDN for the ICT infrastructure of power grids and studied its performance in comparison to MPLS using simulation [6]. In [7] the role of SDN in the creation of self-configuring substations is analyzed. Another glimpse on the several capabilities of SDN-based networks is given in [8], demonstrating a simple approach to traffic engineering for phasor measurement unit traffic. In comparison, several works have elaborated on fault tolerance mechanisms using SDN. First efforts of achieving fault tolerance in OF networks have been undertaken in [9] by introducing BFD and MPLS-based local protection schemes prior to its official integration in OF version 1.1. A similar concept of local failure recovery has been explored in [10] by installing flow-table entries for back-up paths previous to a fault. In [11] restoration and protection mechanisms for the recovery of an OpenFlow-based carrier grade network are compared using the network emulation tool Mininet. Adrichem et al. [12] provide probably the fastest recovery times (about $3\,\mathrm{ms}$) by using OpenFlow's FFG in conjunction with BFD and protecting individual paths in a manner similar to this paper. In [13] a controller-driven restoration mechanism is proposed, which reduces control network load at the expense of recovery delays, especially in larger networks. Yet, none of the above works studies Smart Grid traffic recovery and post-optimization of restored traffic flows is still an open issue of research.

III. Failure Detection and Recovery Concepts

Within this section, the basic concepts and their realization in our SDN for Smart Grids testbed are described. We distinguish three different phases of handling link failures: failure detection, recovery and finally post recovery optimization.

A. Communication Link Failure Detection

Failure detection refers to the methods, applied to identify a fault within the network. In previous work [4], we were able to show that this phase is critical for timely recovery of real-time traffic, since conventional failure detection induces additional delay of about 250 to $350\,\mathrm{ms}$. Therefore, reducing failure detection delay is one major goal of this work. We employ two different approaches: local, autonomous failure detection using Bidirectional Forwarding Detection (BFD) [14] and heartbeat (HB) messages, send out by the SDN controller.

1) Bidirectional Forwarding Detection: BFD is a protocol for local link status monitoring between two switches, proposed by Juniper Networks in 2010, which is included in Open vSwitch [15] as of version 2.3.0. BFD uses network layer protocol specific, lightweight messages (UDP in case of IP, packet size: $70\,\mathrm{Byte}$), which are sent out by the switches at both ends of a link to check a link's liveliness at a minimum inter transmission time (ITT) of $1\,\mathrm{ms}$. The ITT is link specific and negotiated between the two switches, selecting the higher interval to prevent weaker participants from being overloaded with messages. A link is assumed to have failed if no message is received within a multiple of the ITT (common multiplier: 3). We enhanced the SDN controller by enabling remote configuration of BFD on the switches.

2) Controller Heartbeat: In contrast to BFD, our heartbeat messages follow a centralized approach, being sent out by the SDN controller itself. Thus, link failures are detected by the controller directly instead of being informed by the switches. The controller heartbeat uses lightweight Ethernet packets (size: $64\,\mathrm{Byte}$), which are encapsulated in OFPacketOut messages and sent to the switches, adjacent to the monitored link. The switch decapsulates the message and transmits the raw Ethernet packet to the other end of the link, where the receiving switch re-encapsulates it and loops it back to the controller as an OFPacketIn message. Like BFD, the controller checks link status at a multiple of the ITT. Due to small packet sizes, the heartbeat approach scales quite well even for large and densely meshed networks (encapsulated OFPacketOut $p_{po} = 132\,\mathrm{Byte}$, OFPacketIn $p_{pi} = 168\,\mathrm{Byte}$). In a real world scenario, management of $n = 1000$ links by an appropriately designed controller platform, connected via a $r_{cn} = 100\,\mathrm{Gbps}$ port, would cause a control network load of:

$$\eta = \frac{p_{pi}}{ITT} \cdot \frac{n}{r_{cn}} = 1.3\,\% \tag{1}$$

with $ITT = 3\,\mathrm{ms}$. In contrast, BFD would utilize $0.5\,\%$ only.

B. Failure Recovery

Analogous to failure detection, we present two concepts for recovery, a local one (protection) and a controller-driven (restoration). In both cases, packets under way - except for those on the failed link - are re-routed to their destination.

1) Fast Failover Groups: The local recovery approach makes use of OpenFlow's Fast Failover Groups (FFG). The FFG is an OF Group that interconnects a number of action buckets, each consisting of an action and associated parameters. Here, a bucket's action is forwarding incoming packets on a defined output port, while the additional parameter specifies a watch port, whose liveliness is checked. In order to protect a traffic flow, the SDN controller installs FFG at the switches along with regular flow table entries, which reference the groups. Thus, on arrival of a packet, matching the flow table entry, the packet is forwarded to the corresponding FFG, which then executes the action of its first live bucket. However, the major effort in protecting a traffic flow takes place previous to FFG installation by determining an appropriate alternative path. Here, we apply a variation of the Depth First Search (DFS) algorithm to the network topology in order to gain all valid, loop-free paths between two nodes. For each possible link failure, the path with the highest similarity to the original, i.e. optimal, path is selected from the set of feasible paths. In this case, highest similarity of paths refers to the idea that the number of flow table entries necessary for protection should be kept to a minimum, ensuring low memory allocation at the switches. Subsequently, the GroupMod and FlowMod messages for establishing this alternative path are created and written out to the corresponding switches. FFG recovery - respectively the status change of a watch port - might be triggered either by conventional failure detection or by BFD.

2) Controller-driven Recovery: In contrast to the FFG approach, our controller-driven recovery concept does not pre-calculate alternative paths, but looks up the new optimal path after a link failure. Thus, in case the controller is notified of a fault, the affected flows are identified. Next, a valid alternative path is determined, excluding paths which involve the failed link and considering the current network state. Finally, corresponding FlowMod messages are sent to the switches. Controller-driven recovery is triggered by the time-out of a heartbeat or the reception of an OFPortDown message.

C. Post Recovery Optimization for QoS Preservation

The process of post recovery optimization is necessary for establishing new optimal routes after traffic has been recovered with the help of FFG. As described above, FFG recovery does not provide alternative paths, which are optimal with regard to traffic flow requirements but involve minimal overhead at the switches. Thus, the resulting routes might not be suitable for the recovered traffic flows, or worse could even cause overload situations on certain links. Usually, post recovery optimization is not an issue for controller-driven recovery, since optimal routes - avoiding overloaded links - can already be selected during initial recovery. The procedure involves the following steps: traffic flows, affected by the failure are identified and each flow - ordered by priority - is re-processed by our general routing module. Hence, traffic flows with the highest priority are assigned new optimal paths first, while reducing further re-routing efforts. Accordingly, only low priority flows, congesting the new path of high priority traffic need to be moved. In addition, overloaded links are prevented, since the current load situation is considered during path selection already. Finally, the new configuration is published to the switches, intermediate recovery configurations are cleaned-up and new protection paths are established. Thus, strict QoS requirements of Smart Grid applications are met at all times.

IV. Test Environment

The test environment comprises a total of 11 Ethernet (IEEE 802.3ab 1000Base-T) switches, six host computers and one SDN controller. These components are arranged to form three separate networks: data, control and management networks. The data network consists of the host computers for traffic generation and 9 switches, 4 of which are virtual switches running on standard computing hardware, while the other 5 switches are Pica8 3290 bare metal switches. The virtual switches include one onboard Intel I217-LM (control network) as well as a 4 Port Intel I350 Ethernet Network Interface Card (data network) and run Open vSwitch version 2.4.0 as switching software on an Ubuntu Server 14.04.3 64Bit (3.13.0-32-generic Kernel). PicOS 2.6.32 is used on the bare metal switches, featuring Open vSwitch version 2.3.0. The SDN control network is formed by the SDN controller and one Zyxel GS1900-24E switch, interfacing with the data network switches on one dedicated port each. Our SDN controller has been developed as a fork of Floodlight version 1.0 [16], using OpenFlow version 1.3 as southbound protocol. For traffic generation we use the open-source software packETH, enabling the transmission of IEC 61850 specific messages, generated from packet captures, at user-defined intervals.

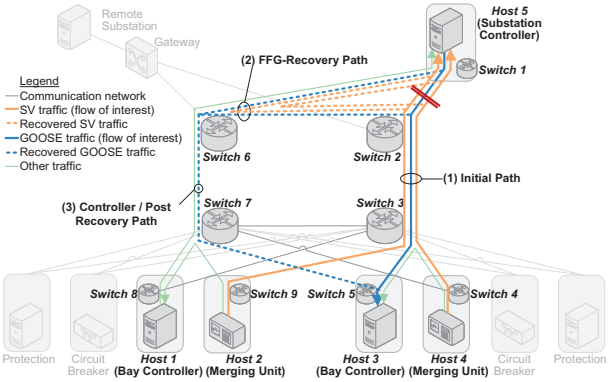

Figure 2: Scenario 1: Two-Bay IEC 61850 Substation Topology with SV, GOOSE Traffic Flows and Recovery Paths

V. Scenarios for Reliability Assessment

This section introduces two Smart Grid scenarios, which are used to evaluate the applicability of the proposed fast failover mechanisms for such critical infrastructures.

A. Substation Automation in Smart Grids

Figure 2 illustrates the communication network for an exemplary IEC 61850 substation with two bays. Each bay consists of the following devices: a bay controller, a protection device, a merging unit, which samples and forwards measurement values as well as a circuit breaker for disconnecting power lines. Moreover, the substation comprises an overall controller and a gateway for connecting to the wide area network. In order to provide a more compact evaluation environment, we exclude protection and circuit breakers from the substation set-up. The communication network is laid out in a ring topology with redundant connections to two different switches for each device. Sampled Value (SV) messages are used for regular transmission of measurement data from both merging units to their respective bay controllers and the substation controller at frequencies of 2000 pps. Commands from the substation controller to both bay controllers are sent as Generic Object Oriented Substation Event (GOOSE) messages. In a real-world scenario such messages occur randomly, triggered by events. Yet, for ensuring reproducibility and mitigating the impact of the ITT on detection times, we apply a fixed ITT of 500 μs. Both, SV and GOOSE, are time-critical IEC 61850 services, encapsulating their payload into Ethernet packets directly.

B. Wide Area Monitoring, Protection and Control

For analyzing wide area communication between substations of the power grid, a section of the IEEE 39 bus reference system (New England Test System) is modeled in the testbed, as illustrated in Figure 3. We assume fibre cables to be carried along with the power lines. To provide redundancy for failure recovery, this communication network has been

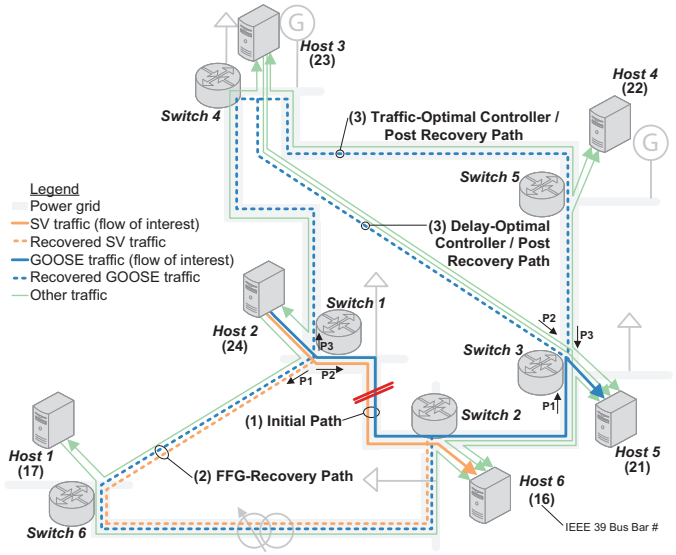

Figure 3: Scenario 2: IEEE 39 Bus Reference System Section with SV, GOOSE Traffic and Respective Recovery Paths

extended with additional connections (substations 17-24 and 21-23). Figure 3 shows the substation numbers along with the corresponding hosts of our testbed. As for the traffic in this scenario, measurement values are exchanged between all neighbouring substations (sharing common power lines) by means of the SV service. GOOSE messages are applied for wide area protection (e.g. differential line protection, distance protection) and remote control of substations. Here, GOOSE packets are sent from host 2 to hosts 1 and 5 and from host 3 to hosts 5 and 6. Due to our SDN infrastructure, GOOSE and SV messages can be routed through the wide area network without further measures such as tunnelling. Yet, additional security provisions need to be considered for real-world deployments.

VI. Evaluation Results

In this section we analyze the performance of our proposed fast failure recovery mechanisms, grouped by scenario.

A. Fast Failover in Substation Automation

For analysing recovery behaviour in the substation scenario we disconnect the link between switch 1 and 2 of the substation infrastructure (c.f. Figure 2). This impacts SV transmissions from hosts 2 and 4 (merging units) to host 5 (substation controller) as well as the GOOSE messages sent from host 5 to host 3 (bay controller). Physical disconnection is required to assess failure detection time, since detection is instant in case a network interface is taken down by command. Figure 4 shows traffic disruption times at the receiver side. Results are grouped by traffic flow and illustrated by means of box an violin plots, comprising measurement values' frequency.
As for BFD we set the ITT to 1 ms with a detect multiplier of 3. Using this configuration in combination with FFG results in traffic recovery delays in the range of 2.39 to 6.62 ms with medians from 3.93 to 4.36 (c.f. Figure 4). Standard deviation varies between 0.64 and 0.75 ms. In contrast, heartbeat detection achieves stable operation not until an ITT of 3 ms and a detection multiplier of 5. Accordingly, the central recovery mechanism results in higher failover delays at the receiver ranging from 14.33 to 24.49 ms, when aggregating over all three traffic flows affected. Here, the median varies between 19.59 and 21.26 ms with the violin plots indicating a wider distribution of recovery times and a standard deviation of 1.58 to 1.92 ms. However, regarding path optimality, the central approach shows better results. For example GOOSE traffic recovers to the path 1-6-2-3-5 in case of local recovery, requiring one more hop than the initial path. In contrast central recovery selects the path 1-6-7-5. This behaviour is a result of FFG redirecting traffic to the original path as soon as possible to avoid massive switch flow table overhead.

B. Fast Failover in Wide Area Monitoring, Protection and Control

In the second scenario, we fail the link between switch 1 and 2, disrupting the exchange of SV between host 2 (substation 24) and host 6 (substation 16) as well as the transmission of GOOSE messages from host 2 to 5 (substation 21). Local recovery redirects traffic to switch-path 1-6-2 to

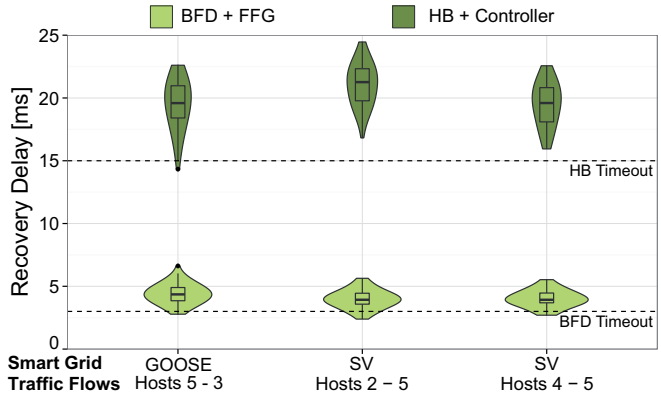

Figure 4: Failover Times using Local and Central Recovery Mechanisms in Scenario 1, Measured at the Receiver

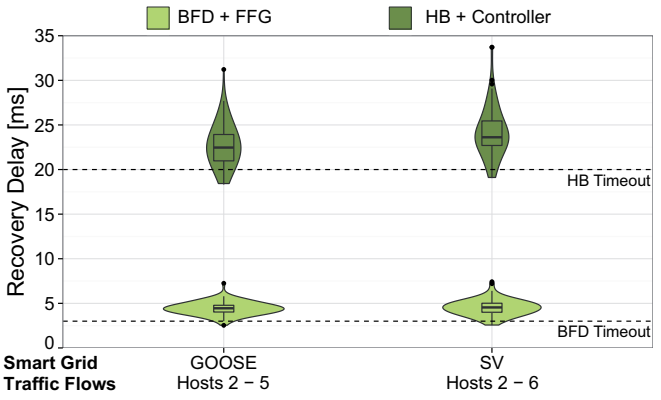

Figure 5: Failover Times using Local and Central Recovery Mechanisms in Scenario 2, Measured at the Receiver

enable immediate return to the original path. While this path is optimal for SV traffic, it results in a suboptimal route for GOOSE traffic (switches 1-6-2-3). In comparison, the central mechanism establishes new routes for GOOSE traffic either traversing switches 1-4-3 or 1-4-5-3, depending on the routing policy selected. On the other hand BFD and FFG achieve significantly faster recovery with a median of $4.44\,\mathrm{ms}$ for GOOSE and $4.54\,\mathrm{ms}$ for SV traffic (standard deviation 0.71 respectively $0.84\,\mathrm{ms}$), applying the minimum ITT of $1\,\mathrm{ms}$ and 3 as detect multiplier. The controller-based heartbeat mechanisms, stabilizes not until an ITT of $5\,\mathrm{ms}$ and a detect multiplier of 4. This configuration results in traffic recovery times at the receiver side from 18.32 to $33.71\,\mathrm{ms}$. GOOSE traffic shows a median disruption time of $22.46\,\mathrm{ms}$ with a standard deviation $2.29\,\mathrm{ms}$, while SV messages exhibit a median recovery duration of $23.62\,\mathrm{ms}$ and a standard deviation of $2.73\,\mathrm{ms}$. Again, this emphasizes better performance of the local mechanism with regard to recovery times and their corresponding standard deviation. Timing issues of the heartbeat mechanism can be attributed to its implementation in JAVA, which is due to direct integration into our JAVA-based controller framework. Performance might be improved by outsourcing send-/receive processes into C code.

C. Hybrid Failover Approach for Post Recovery Optimization

To compensate the shortcomings of the two individual approaches, we propose the combination of local and central recovery mechanisms into a hybrid one. Immediate failover is achieved using BFD in conjunction with FFG to minimize disruption times. Afterwards, new QoS optimal paths are computed and pushed to the switches by the controller. Here, the controller detects failures actively via its heartbeat mechanism. Figures 6 and 8 show examples of complete failover processes including post recovery optimization for the WAMPAC scenario using two alternative routing policies. The procedure is visualized by capturing the traffic load on the network interfaces of switches 1 and 3 (c.f. Figure 3).

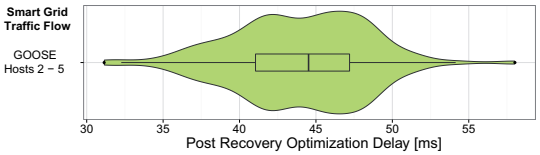

Figure 7: Optimization Delay for Hybrid Failover Approach

Applying a delay optimizing routing policy GOOSE traffic is switched from the fast failover path via switches 1-6-2-3 to the optimized recovery path (switches 1-4-3). This can be deduced from traffic migrating from port 2 to 3 at switch 1 respectively from port 1 to 2 at switch 3 in Figure 6. Previously, during fast failover, traffic is redirected from port 1 to 2 at switch 1, while it remains on port 2 at switch 3. In contrast, if routing is focussed on load balancing, traffic is switched to the optimized path via switches 1-4-5-3, as evident by the traffic at switch 3 migrating from port 1 to 3 in Figure 8. Here, this specific route is chosen since both other feasible paths (switches 1-6-2-3, 1-4-3) are utilized by two GOOSE traffic flows respectively one GOOSE and one SV traffic flow. Meanwhile, the fast failover route is identical, regardless of the routing policy. Figure 7 illustrates the delay between failure occurrence and recovery to a new optimal path, accumulating the results of both routing policies. Aggregation is possible without loss of accuracy, since 1) only a minor part of the delay (lower microsecond range) is incurred by the controller at all [4] and 2) different routing policies result in minimal deviations in delay in this scenario. For the heartbeat, we select an ITT of $10\,\mathrm{ms}$ and 4 as detect multiplier to achieve stable operation at all times. Accordingly, post recovery delay exhibits a mean value of $43.98\,\mathrm{ms}$ and a standard deviation of $4.41\,\mathrm{ms}$, measured at switch 3. While limited timing precision of the heartbeat is negligible for post recovery optimization, combined usage of two detection mechanisms generates additional overhead. This might be optimized by using BFD to trigger OFPortStatus messages to the controller.

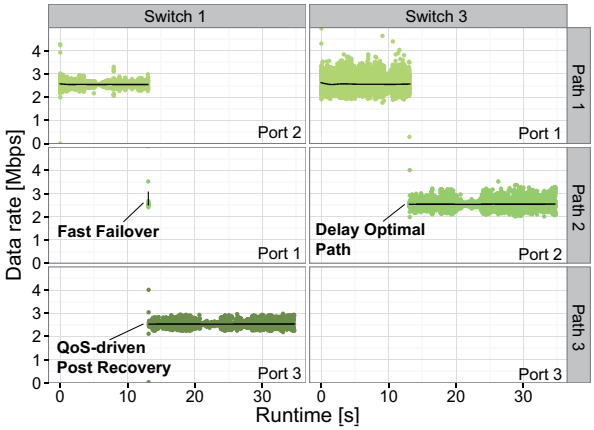

Figure 6: Delay Optimized Routing Policy: Complete Recovery Process at Switches 1 and 3, Illustrating Smart Grid Traffic Migration between Different Network Interfaces (Ports)

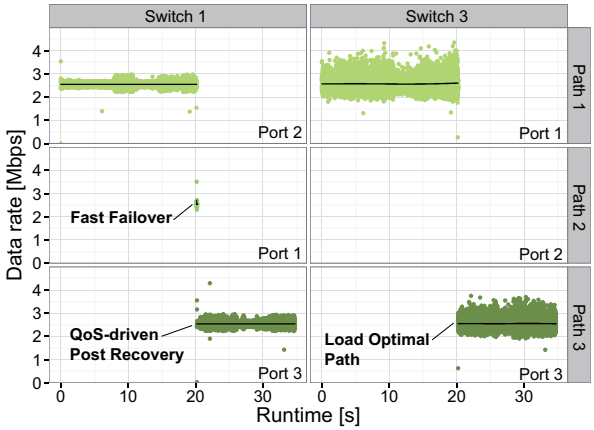

Figure 8: Load Balanced Routing Policy: Complete Recovery Process at Switches 1 and 3, Illustrating Smart Grid Traffic Migration between Different Network Interfaces (Ports)

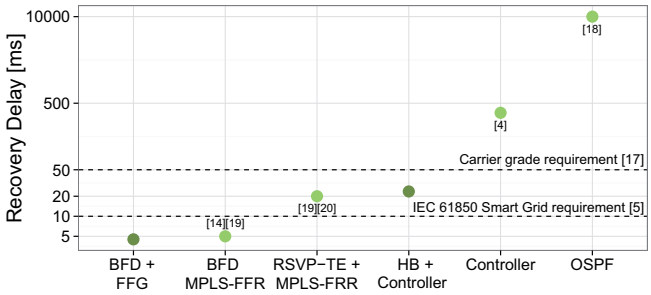

Figure 9: Comparison of the Proposed SDN-based Fast Failover Approaches with Traditional Recovery Mechanisms

VII. Conclusion

In this work we realize and analyse local as well as central mechanisms for fast failover in software-defined Smart Grid communication networks. Table I shows a qualitative summary of our results, indicating supremacy of the local approach with regard to recovery time and scalability. In contrast, the controller-driven detection and recovery concept provides benefits in terms of path optimality due to the controller's global network view. The proposed hybrid solution shows promising results, combining advantages of local and central approach, yet at the expense of additional overhead. Hence, fast recovery of critical Smart Grid traffic is achieved, respecting additional QoS requirements such as latency and traffic priority.

Table I: Qualitative Comparison of Local, Central and Hybrid Failure Recovery Approaches

Approach	Rec. Time	Scalability	Path Optimality	Overhead
BFD + FFG	+	+	-	+
HB + Controller	○	-	+	+
Hybrid	+	○	+	○

Complementary to this summary, Figure 9 provides a comparison of the proposed OpenFlow-reliant recovery mechanism with traditional approaches like Open Shortest Path First (OSPF) or MPLS with either BFD or Resource Reservation Protocol - Traffic Engineering (RSVP-TE) Hello for detection. Both of our methods are able to compete with well-established mechanisms and are capable of fulfilling the sub 50 ms requirement of carrier grade networks [17] on average. Using BFD for failure detection, fast recovery in software-defined networks keeps up with the combination of BFD and Fast Reroute (FRR) in MPLS-based infrastructures. SDN-based local and hybrid recovery mechanisms satisfy even more strict demands of the IEC 61850 substation environment (sub 10 ms) [5], proving their suitability for such critical applications. Future work will enhance our hybrid recovery approach to reduce overhead as detailed in Section VI-C. Besides, we plan on further extending our prioritisation methods and providing a northbound interface at the SDN controller for interacting with Smart Grid applications.

Acknowledgement

This work has been carried out in the course of research unit 1511 *'Protection and control systems for reliable and secure operations of electrical transmission systems'*, funded by the German Research Foundation (DFG) and the Franco-German Project *BERCOM* (FKZ: 13N13741) co-funded by the German Federal Ministry of Education and Research (BMBF).

References

[1] Y. Yan, Y. Qian, H. Sharif and D. Tipper, 'A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges', *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 5–20, 2013.

[2] N. McKeown *et al.*, 'OpenFlow: Enabling Innovation in Campus Networks', *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[3] Open Networking Foundation, *OpenFlow Switch Specification Version 1.3.0*, 2012. [Online]. Available: https://www.opennetworking.org/.

[4] N. Dorsch, F. Kurtz, H. Georg, C. Hägerling and C. Wietfeld, 'Software-Defined Networking for Smart Grid Communications: Applications, Challenges and Advantages', in *IEEE International Conference on Smart Grid Communications*, Nov. 2014, pp. 422–427.

[5] IEC TC57, *IEC 61850: Communication Networks and Systems for Power Utility Automation*, International Electrotechnical Commission.

[6] A. Sydney, J. Nutaro, C. Scoglio, D. Gruenbacher and N. Schulz, 'Simulative Comparison of Multiprotocol Label Switching and OpenFlow Network Technologies for Transmission Operations', *Transactions on Smart Grids*, vol. 4, no. 2, pp. 763–770, 2013.

[7] A. Cahn, J. Hoyos, M. Hulse and E. Keller, 'Software-Defined Energy Communication Networks: From Substation Automation to Future Smart Grids', in *IEEE International Conference on Smart Grid Communications*, Toronto, Canada, 2013, pp. 558–563.

[8] A. Goodney, S. Kumar, A. Ravi and Y. H. Cho, 'Efficient PMU Networking with Software Defined Networks', in *IEEE International Conference on Smart Grid Communications*, Toronto, Canada, 2013, pp. 378–383.

[9] J. Kempf *et al.*, 'Scalable fault management for OpenFlow', in *IEEE International Conference on Communications*, Jun. 2012, pp. 6606–6610.

[10] A. Sgambelluri, A. Giorgetti, F. Cugini, F. Paolucci and P. Castoldi, 'OpenFlow-based segment protection in Ethernet networks', *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 9, pp. 1066–1075, Sep. 2013.

[11] S. Sharma, D. Staessens, D. Colle, M. Pickavet and P. Demeester, 'Fast failure recovery for in-band OpenFlow networks', in *International Conference on the Design of Reliable Communication Networks*, Mar. 2013, pp. 52–59.

[12] N. van Adrichem, B. Van Asten and F. Kuipers, 'Fast Recovery in Software-Defined Networks', in *European Workshop on Software Defined Networks (EWSDN)*, Sep. 2014, pp. 61–66.

[13] S. S. W. Lee, K. Y. Li, K. Y. Chan, G. H. Lai and Y. C. Chung, 'Software-based fast failure recovery for resilient OpenFlow networks', in *International Workshop on Reliable Networks Design and Modeling*, Oct. 2015, pp. 194–200.

[14] D. Katz and D. Ward, *Bidirectional Forwarding Detection (BFD) (RFC 5880)*, Internet Engineering Task Force (IETF), Jun. 2010.

[15] *OpenFlow vSwitch Version 2.4.0/2.3.0*, 2015. [Online]. Available: http://openvswitch.org/.

[16] Project Floodlight, *Floodlight Controller Version 1.0*, 2015. [Online]. Available: http://www.projectfloodlight.org/floodlight/.

[17] B. Niven-Jenkins, D. Brugard, M. Betts, N. Sprecher and S. Ueno, *Requirements of an MPLS Transport Profile (RFC 5654)*, Internet Engineering Task Force, Sep. 2009.

[18] J. Moy, *OSPF Version 2 (RFC 2328)*, Internet Engineering Task Force, Apr. 1998.

[19] Y. Lei, C.-H. Lung and A. Srinivasan, 'A Cost-Effective Protection and Restoration Mechanism for Ethernet Based Networks: an Experiment Report', in *Workshop on High Performance Switching and Routing*, 2004, pp. 350–354.

[20] O. Komolafe and J. Sventek, 'Analysis of RSVP-TE Graceful Restart', in *IEEE International Conference on Communications*, Jun. 2007, pp. 2324–2329.