Layer-2 Failure Recovery Methods in Critical Communication Networks

Ferdinand von Tüllenburg and Thomas Pfeiffenberger

Salzburg Research Forschungsgesellschaft mbH
Advanced Networking Center
email: ferdinand.tuellenburg@salzburgresearch.at
email: thomas.pfeiffenberger@salzburgresearch.at

***Abstract*—Service interruptions in critical infrastructures, like the power grid, can lead to serious consequences for safety and security of people. To avoid such interruptions of distributed applications or process control systems belonging to a critical infrastructure, reliable recovery mechanisms for the associated communication systems are essential. OpenFlow, a standard for software defined networking (SDN), provides the fast failover group mechanism to forward packets via alternative paths in case of link failures. In contrast to the conceptual and theoretical discussions of this concept, in this work, the performance of path restoration using SDN fast-failover groups is compared to the performance of path computation when using the Rapid Spanning Tree Protocol (RSTP). Our results show, that current implementations of OpenFlow can significantly improve the failover performance compared to RSTP, which makes it possible to use SDN in ultra high reliability communication networks. But it is also shown that there is a potential to further improve the SDN recovery mechanisms by deeply inspecting the correlations between OpenFlow/SDN implementations, the used hardware and the operating system.**

***Index Terms*—Critical infrastructure; Fast failover evaluation; Software defined networking; Reliability; Network recovery**

I. Introduction

Some technical systems like the electrical grid or other utility systems are of special importance for our modern society as they are providing the basis on which our communities, economies and everyday lives are founded. Such technical systems are referred to as critical infrastructures. In the last years, there has been a recognizable trend of a proceeding augmentation with Information and Communication Technology (ICT) to increase advantages and efficiency of such systems. With this progress, critical infrastructures are getting highly dependent on a working communication infrastructure, making this ICT itself to a critical infrastructure [1]. One example for this development can be seen in case of power grids, which are evolving towards Smart Grids. Here, various entities of the power systems like generators, sensors, and intelligent devices are getting interconnected using ICT in order to enrich the power grid with more sophisticated functions for monitoring and control, trading, and Demand Side Management [2].

Nowadays, communication networks for critical infrastructures are often operated as dedicated networks where connections to other networks (especially the Internet) are avoided. Mainly due to the risk of introducing security and performance issues as certain ICT functions in critical infrastructures have special requirements for reliability, data security and quality of service (QoS). This however, has several disadvantages such as high operating and installation costs for dedicated networks and the impossibility to share information between systems belonging to different critical infrastructures. But, when already existing communication infrastructures are extended to be used for critical infrastructures beside its originary operation purpose, methods are needed that guarantee the reliability of critical traffic. To achieve this, (1) critical traffic should be separated from non-critical traffic and (2) for critical traffic special treatment is needed in order to guarantee communication reliability. One approach often discussed in current and recent research project is the use of Multiprotocol Label Switching (MPLS) networks. With MPLS and its traffic engineering extension Ressource Reservation Protocol - Traffic Engineering (RSVP-TE), traffic of distinct applications can be forwarded differently within the network, which leads to traffic separation. Furthermore, with RSVP fast reroute a fast method is given to reroute packets as soon as link failures occur. This can be used to increase the communication reliability of critical traffic. The disadvantage of MPLS, however, are high efforts for maintenance and often high costs for provisioning of MPLS services (e.g., renting MPLS lines from service providers).

Software-defined networking (SDN) provides other approaches to tackle the aforementioned topic. This can be shown in the SDN testbed for critical and non-critical applications. Here, SDN is considered as a promising candidate to separate traffic of critical and non-critical applications. This also goes in conjunction with better mitigation of potential security risks, increased reliability through isolation from configuration errors of other applications and networks, and a simplified configuration and management for both, infrastructure users and providers.

The SDN testbed implements solutions as a proof of concept in a real-end user, OpenFlow enabled [3] fibre to the home infrastructure operated by a district heat provider. The district

 ISBN: 978-1-61208-482-4

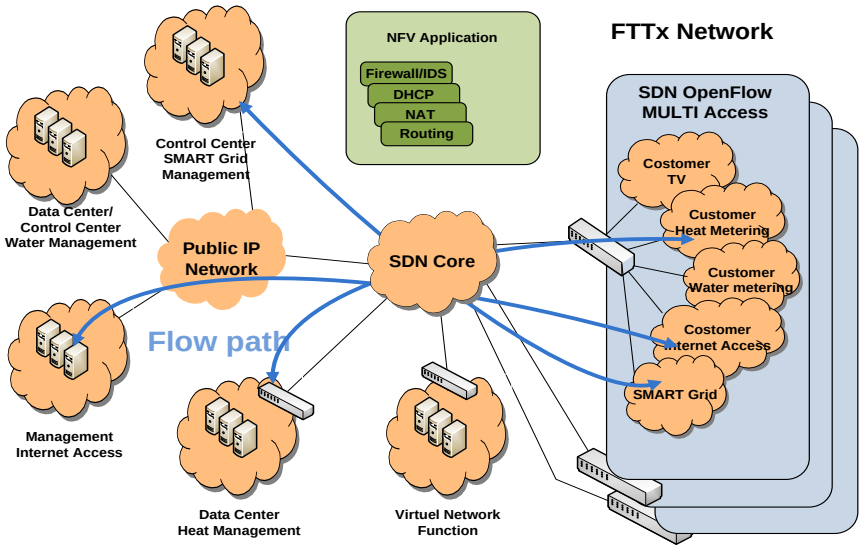

Figure 1. SDN testbed for co-existence of critical and non-critical network applications.

heat provider uses this infrastructure on the one hand for controlling purposes of the heating system, and on the other hand to offer his communication infrastructure to service operators, which in turn offer additional services (such as high speed Internet) to customers. Also the integration of metering solution for smart grids or water systems for utility service providers or the local government is possible. Figure 1 gives an overview of the SDN based testbed.

The topic addressed in the current study is reliability of critical infrastructure communication. Certain applications of critical infrastructures require a high degree of reliability regarding communication interruptions. Here, OpenFlow provides a special fast failure recovery method allowing for configuring switches at the SDN forwarding plane with fast-failover groups. Such a group defines a list of alternative ports on which packets can be sent out belonging to a certain traffic trunk. This means, that for all computed paths between a source and a destination, all switches on these paths can have multiple opportunities to forward packets to follow one of the precomputed paths. As soon as one forwarding port is not usable in case of a link failure the alternative port can be used. The decision, which one of the possible output ports is used, is taken at each switch based on the locally available link state information.

In the present performance comparison study, OpenFlow fast-failover is compared to the commonly used rapid spanning tree protocol (RSTP) that also provides the reroute capabilities when link failures occur. In difference to OpenFlow's failover groups, alternative paths are computed by a distributed algorithm right after a link failure occurs [4]. In this paper, we focus on the comparison of the OpenFlow fast failover groups with RSTP as we had a focus on layer 2 of the OSI model. Further more the study is done to examine the applicability of SDN/OpenFlow for reliable communication in the SDN Testbed. The aim of this work is also to encourage further discussions on augmenting communication networks of critical infrastructures with SDN technology.

The paper is organized as follows: Section II contains a brief overview of other work related to this paper. In Section III a short introduction to MPLS fast reroute method is depicted. Section IV describes the network infrastructure used for the tests as well as the methodology of the tests. Section V describes the validation results in detail before we give an outlook on future work in Section VI.

II. Related Work

Several studies have investigated the application of SDN in Smart Grid communication systems. Dong et al. focus on possibilities of SDN to improve the resilience of a Smart Grid. They also discuss critical issues of SDN, which need to be taken into account before deploying SDN to Smart Grids [5]. The use of SDN in the area of substation automation based on IEC-61850 is discussed in [6]. Here, Cahn et al. describe a system that automatically configures the network infrastructure of a substation with respect to the communication requirements of present IEDs and monitoring devices. This work is brought to a more practical level in [7], where the current development state of SDN/OpenFlow implementations was investigated in detail and, in addition, the ability of SDN to fulfill communication requirements of Smart Grid communication networks was evaluated. [8], [9] and [10] evaluates different methodologies to implement failure recovery in SDN based networks. Dorsch et al proposed approaches for fast-recovery and guaranteed quality of service [2]. In contrast to the fast-recovery approach proposed in our work, the logic for re-routing of packets is centralized at the SDN Controller - i. e., if a link failure occurs, the corresponding switches send a message to the SDN controller, asking for an alternative forwarding rule. The approach of our work utilizes OpenFlow fast-failover groups to provide multiple alternative paths to

 ISBN: 978-1-61208-482-4

the switches at the same time. Switching to alternative paths can be done based on local decisions of switches, which is expected to reduce link down time and packet loss. Our approach could be extended by the work of [11] where a SDN Controller precalculates multiple forwarding paths from a sender to a destination at the same time, and download the corresponding forwarding rules to the switches. In such an approach OpenFlow enabled switches have to deal with a significant higher number of flow entries and this harbours the risk of flow table explosion.

III. RSVP-TE Fast Reroute

For critical infrastructure communication, MPLS (and especially its extension RSVP-TE) is frequently proposed in order to guarantee reliability, traffic separation, reliable bandwidth separation and the like. RSVP-TE enables to establish label switched paths (LSP) throughout an MPLS network including resource reservation on end-to-end links such as minimum bandwidth or delay requirements. One extension of RSVP-TE to LSPs is fast reroute functionality. Fast reroute allows the establishment of additional backup LSPs which can be switched to as soon a link failure or network failure occurs. RSVP-TE fast reroute is specified in RFC 4090 [12].

In general, fast reroute works according to the following simplified model: When a new LSP is requested (usually by the network administrator), several detours are precomputed and preestablished along the LSP. These detours are paths between MPLS routers, which provide local repair capabilities. After a link failure has been detected by a directly connected Label Switching Router (LSR) it becomes to the point of local repair and uses one of the preestablished detours to quickly reroute traffic around the failure point. In a second step, after rerouting the traffic via the detour, the router sends a notification to the MPLS ingress router, which then, establishes a complete new LSP avoiding the network failure point. While the computation of a new LSP takes several seconds, the local repair can be established within several milliseconds after a failure has been detected.

For the detection of link and node failures, the fast reroute makes use of MPLS hello messages for the detection of unreachable neighbour MPLS nodes and additionally can make use of local physical layer information to detect link failures to next-hop neighbours. While rerouting based on the local link information can be done within some milliseconds, using hello messages to detect failed neighbor nodes takes times in the scale of some seconds. In the latter case, hello messages are periodically (every two to five seconds) sent out by LSRs to their neighbours and if no reply is received from a neighbour LSR, this LSR is considered as broken. Due to this, a link failure can remain undetected several seconds before the local repair mechanism starts to work. In several practical implementations and evaluations, it has been shown that fast reroute can reach failure recovery times in the range of up to 50 milliseconds for the local repair mechanisms when physical layer information is used [13].

While a direct comparison between MPLS fast reroute networks and SDN approaches would be highly interesting, in this paper we are focusing on pure layer 2 link failure recovery techniques. For a comparison of MPLS fast reroute and SDN approaches, a more comprehensive evaluation should be done including also features like bandwidth protection, which are provided by RSVP-TE fast reroute.

IV. Validation Architecture and Methodology

Content of this section is a description of testbed architectures and testing methodologies for the failover performance evaluation of SDN/OpenFlow and RSTP.

A. RSTP fast-failover evaluation

The network infrastructure used for RSTP evaluation consists of two hosts A and B, connected with a network of four switching devices S1, S2, S3, and S4 (see Figure 2).

Both end devices are standard desktop computers using 1 Gbit/s standard Ethernet interface cards. On host A a sending application is able to send UDP flows with a configurable packet size and sending interval. During the evaluation measurements, the packets are forwarded through the network and finally delivered to the destination host B. At host B a receiving application is running, which captures the packets sent by host A and keeps track of receiving timestamps of each packet. The UDP packets' payload containing their sending timestamps and a packet number increased by 1 for each packet sent out (the first packet has number 0). Both contents are written by the sending application. By utilizing the packet numbers stored in the packet's payload, it is possible to compute lost, reordered, and duplicated packets. The receiving application keeps track about the packet numbers of incoming packets. If gaps are detected the according packets are considered as lost. In the case of out-of-order packet numbers of incoming packets, packet reordering is considered. If one packet with the same packet number is received twice, packet duplication can be assumed.

In the RSTP test network, standard desktop computers are used as switching devices running Open vSwitch 2.3.90 supporting RSTP IEEE 802.1D-2004 [4]. All PC based switches

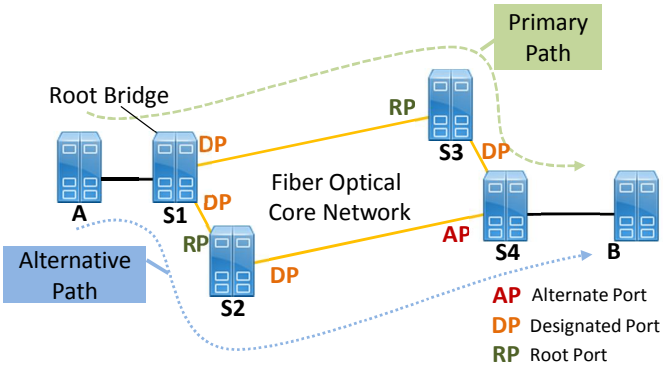

Figure 2. Overview of RSTP evaluation network infrastructure.

TABLE I. OPTIMIZED RSTP PARAMETERS

Parameter	standard value	optimized value
Forwarding Delay	15	4
BPDU max. age	20	6
Transmit Hold Count	6	1
BPDU timeout	1200	1

(S1, S2, S3 and S4) are equipped with identical fiber optical network interface cards in order to make sure that impacts of different network hardware on the recalculation performance can be excluded.

At the beginning of the test, a primary path is computed by RSTP leading via the switches S1, S3, and S4 to the destination host B. When a link failure occurs between S3 and S4, RSTP establishes the alternate path via S2 and S4 to host B (see Figure 2). As soon as the connection between S3 and S4 is available again, the primary path gets restored by the RSTP path computing algorithm.

During the test, the fiber optical connection of the primary path were automatically (by a test program) disconnected and reconnected in time-intervals of 10 seconds. In total 40 disconnect and reconnect actions has been executed during the tests. Each action led to path recalculations of the RSTP protocol in order to find most cost-efficient path towards its neighboring switches and establish a new forwarding path. When the link failure occured (after disconnection of the primary path) RSTP re-established a path via the alternate path and the computation time for the alternate path was recorded. When the broken link has been reactivated, the time needed to return to the default route has been measured.

To maximize the speed of RSTP link failure detection and path calculation, the algorithm parameters forwarding delay, BPDU sending interval, and maximum age of BPDUs are reduced compared to standard values. Table I contains a comparison between RSTP standard values and optimized values.

B. OpenFlow fast-failover evaluation

The network infrastructure for the evaluation of OpenFlow fast-failover performance has the setup shown in Figure 3. The hosts A and B are standard desktop computers, equipped with 1 Gbit/s standard Ethernet network interface cards. These interfaces are faced to the network used for OpenFlow performance evaluation. The switch S1 is a standard desktop PC configured as switch and is running Open vSwitch Version 2.3.90 as Linux Kernel module supporting OpenFlow until Version 1.3. S1 is equipped with a dual port fiber-optical network interface card. One of the ports is connected to switch S2, the other to switch S3. Finally, the switches S2 and S3 are connected to switch S4, which in turn is connected to host B. The switches S2, S3, and S4 are identically built switching devices providing a 1 Gbit/s fiber-optical network. The hardware is natively running Open vSwitch Version 1.9.90, also supporting OpenFlow up to Version 1.3. The switching hardware S2, S3, and S4 are generally i386 Linux boxes with designated TCAM based switching hardware bringing mainly a performance boost for their forwarding actions (TCAM based rule selection).

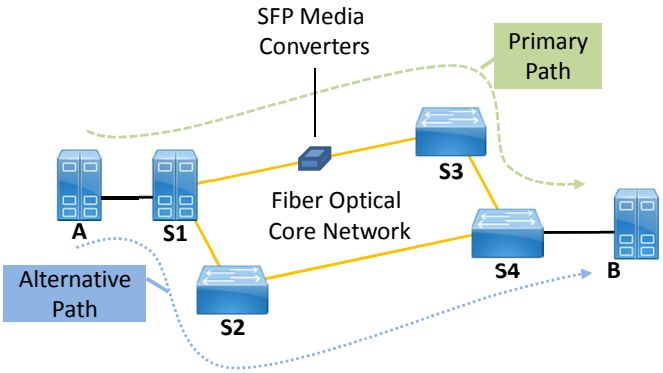

Figure 3. Overview of the OpenFlow/SDN evaluation network infrastructure.

Like in the RSTP tests, on host A, UDP traffic is generated and sent to host B, using the same applications and configurations as in the RSTP evaluation.

To evaluate the OpenFlow fast failover behavior, the network is configured with a fast-failover group at S1, which forwards packets to switch S3 during default operation (primary path). When S1 looses its connection to S3 the alternate path via S2 is used. Furthermore, switch S2 and switch S3 are configured with static flow entries in order to forward packets to switch S4. Switch S4 also has a static flow entry to forward all incoming packets to destination host B.

The sending application is configured to generate Ethernet traffic of about 4.6 MBit/s with following properties:

- 242 Bytes payload of each UDP packet
- 8 Bytes for the UDP header
- 20 Bytes IP header
- 18 Bytes Ethernet header
- 500 microseconds mean packet sending interval

The performance tests were carried out in one automated test scenario to emulate software failures, and one manual test scenario to emulate hardware link failures. This is also done to unveil impacts on the link failure detection mechanism, depending on whether the link failure is produced by turning off the network interface via a user space command or when a physical network connection breaks.

In the automated test scenario, the link between S1 and S2 is interrupted by a disconnection command, which instructs the operating system to deactivate the network interface on switch S1, which is connected to S2. After a waiting time of 10 seconds, the network interface is reactivated again. This procedure is repeated every 10 seconds until the UDP traffic flow from host A to host B has stopped. The automated test is done during a test period of 40 switching actions. The manual test scenario, the link interruption is done manually by interrupting the physical optical fiber connections. Like in the automated scenario, the 10 seconds interval when connecting and disconnecting the link is kept and also 40 switching actions are performed.

 ISBN: 978-1-61208-482-4

In the manual test scenario a SFP-based fiber optical media converter is placed between the switches S1 and S3 (see Figure 3). These devices are normally used, e.g., to convert a single mode fiber optical connection into a multi-mode connection. In the manual test, however, the media converters are used to manually interrupt the connection between S1 and S3. For this purpose the external power connection of the media converter is used. This has the benefit, that contact chatter can be avoided, which was observed when manually plug and unplug optical fibers into the NIC ports. When the media converter is turned off, a link failure appears at both sides of the connections, i. e., both switches S1 and S3 will detect the link failure.

V. Results

In the following section we present and discuss our fast-failover evaluation results with RSTP and OpenFlow.

A. RSTP fast-failover evaluation

When looking at RSTP, the protocol need to recompute the path in two cases: As soon as the primary path fails, which results in a fast transition of the forwarding behavior of switch S4 from its preferred designated port (connected to S3) to its alternate port (connected to S2), and as soon as the primary path has been restored. Then, switch S3 changes to use the more cost efficient (originary) designated port.

For changing the forwarding behavior at switch S4 from the designated port to the alternate port in case of primary link failures, it took 3 ms in minimum and 65 ms in maximum. The average time to establish the backup connection between both hosts were 26 ms. For link reactivations after the primary path has been re-established, RSTP need in minimum of about 500 microseconds and in maximum 809 milliseconds. The mean time for re-establishing the primary path after a reconnect (path restore) is about 401 ms. The differences between first and second case as well as the large interval between minimum and maximum values (in the second case) cannot be conclusively explained but one reason is surely introduced by operating system scheduling and hardware control at the computer hosting switch S4. For the evaluation of these path computation times all 40 connection and disconnection actions has been considered.

One hint in this direction is also given, when comparing our results to those published in a Siemens Whitepaper covering a RSTP performance evaluation [14]. In this work failover times between 50 ms and 100 ms has been measured depending on the networks' size and using specialized RUGGEDCOM switches supporting RSTP standard 802.1D-2004.

B. OpenFlow fast-failover evaluation

The results of the OpenFlow fast-failover evaluation are given by packet losses occurred following a switching action. From the amount of lost packets a worst-case estimation for the interruption time can be made.

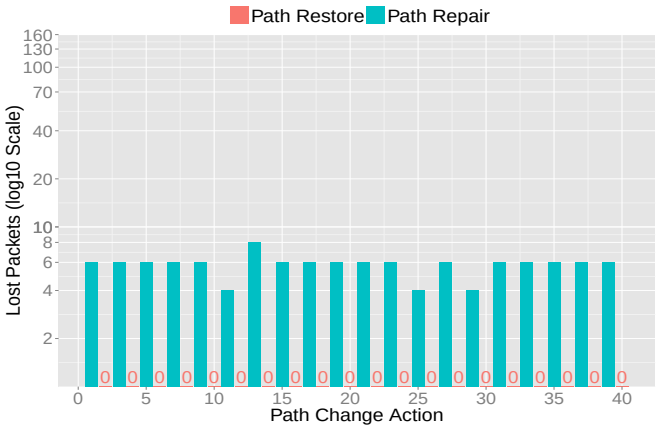

Figure 4. Losses appeared in the automated test scenario.

The bar plots 4 and 5 showing the packet losses resulting from switching actions. On the horizontal axis, the switching actions (Path Change Action) are depicted, on the logarithmic scaled vertical axis, the amount of lost packets per switching action is outlined. All three figures using the same scale for x and y axis. The bars denoted with 'Path Repair' representing packet losses occurred when the primary path in the evaluation architecture is deactivated in a automatic or manual manner. The bars denoted with 'Path Restore' representing packet losses occurred after the primary path was restored and the packet flow was switched back to use the primary path. When no packet losses occurred after a switching action the symbol '0' is written to the plot.

Figure 4 shows the packet losses in case of the automated test scenario when link interruptions were initiated by software. As can be seen from the plots, no packet losses occurred when packet forwarding was switched from alternate to primary path ('Path Restore'). When looking at packet losses occurred after the primary path was interrupted, the mean amount of packet losses was 5.8 packets (minimum 4 packets, maximum 8 packets). As it can be seen from the figure, mostly, the amount of lost packets were 6. This results in a estimated worst-case interruption time between 3 ms and 5 ms (considering a mean packet sending interval of 500 microseconds).

Figure 5 shows the packet losses in case of the manual test scenario with 1 converter, and the link between switch S1 and switch S3 is physically disconnected. Looking at the 'Path Restore' bars, packet loss after switching from the alternate path to the primary path occurred only once. Here, 18 packets got lost meaning a worst-case interruption time of 10 ms. When looking at packet losses occurred after the primary path was interrupted, the mean amount of packet losses was 32.45 packets. The minimum of lost packets was 2, and the maximum 160. As the median of the amount of lost packets were 4, it can be seen that high loss values are not frequent. This also is shown by the 3rd quantile of the measured values lying at 28.00 packets. Considering the

 ISBN: 978-1-61208-482-4

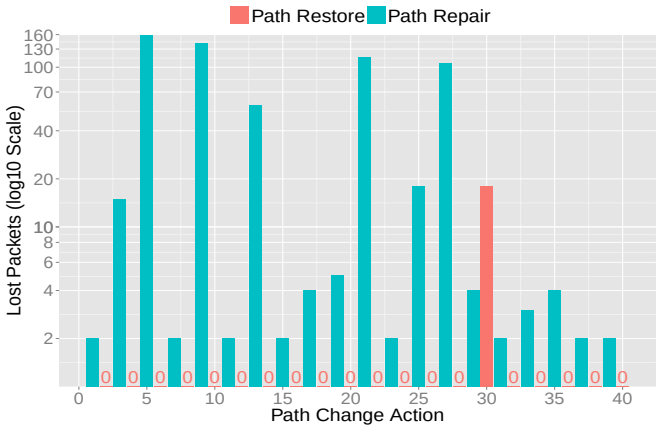

Figure 5. Losses appeared in manual test scenario 1.

mean amount of lost packets, the worst-case interruption time was around 17 ms. Taking the maximum packet loss as a basis, the worst-case interruption time is around 81 ms. For the worst-case estimation, again, a packet sending interval of 500 microseconds is considered.

Comparing the OpenFlow fast-failover scenarios, a very important result is that in most of the cases, no packet losses occurred when switching back from the alternate path to the primary path. Furthermore, the amount of packet losses and interruption has a larger spread and variability in case of the manual test scenario. Comparing the results of RSTP and OpenFlow/SDN, the fast-failover performance using SDN/OpenFlow is significantly improved. While the maximum time for switching a path with RSTP was measured with about 800 ms (mean: 200 ms) in our measurements (Siemens measured with specialized Hardware up to 100 ms) with OpenFlow we measured a maximum interruption time around 81 ms (mean: 17 ms).

VI. Conclusion and Future Work

Our results show that using OpenFlow fast-failover recovery clearly outperforms path recovery mechanisms of RSTP. This makes SDN/OpenFlow a promising approach for establishing an ultra-high reliable communication network for critical infrastructures. A further aspect is that OpenFlow based fast-failover mechanism is able to manage a symmetric loss and timing behaviour. RSTP uses different port roles and port states and this result in an asymmetric behaviour.

On the other hand, our results showed, that the performance of SDN fast-failover highly correlates with used Open vSwitch implementations, networking hardware and operating system support. To fully understand these correlations, it is necessary to extend our validation methodology, e.g., to be able to measure the packet one way delay accurately in the range of small fragments of microseconds. Possibly, clock synchronization using PTP or Sync-E could be used. A further question we want to answer is, how hardware based and high performance network stacks and package processing approaches like DPDK [15] or OpenOnload [16] can improve the quality of failure resistance of SDN/OpenFlow networks.

Answering these questions can lead to high quality productive networks usable for critical infrastructures and other domains like industrial automation systems where latency in the range of micro- and nanoseconds is requested.

VII. Acknowledgment

The work described in this paper was funded by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT).

References

[1] Council of the European Union, "Council Directive 2008/114/EC of 8 December 2008 on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection," *Official Journal of the European Union*, vol. L 345, pp. p. 75 – 82, 2008.

[2] N. Dorsch, F. Kurtz, H. Georg, C. Hagerling, and C. Wietfeld, "Software-defined networking for Smart Grid communications: Applications, challenges and advantages," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. IEEE, Nov. 2014, pp. 422–427.

[3] Open Networking Foundation, "OpenFlow Switch Specification 1.4.0," Oct. 2013, technical Specification TS-012.

[4] IEEE, "IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges," 2004.

[5] X. Dong, H. Lin, R. Tan, R. K. Iyer, and Z. Kalbarczyk, "Software-Defined Networking for Smart Grid Resilience: Opportunities and Challenges," in *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security (CPSS '15)*. ACM Press, 2015, pp. 61–68.

[6] A. Cahn, J. Hoyos, M. Hulse, and E. Keller, "Software-defined energy communication networks: From substation automation to future smart grids," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*. IEEE, 2013, pp. 558–563.

[7] T. Pfeiffenberger and J. L. Du, "Evaluation of software-defined networking for power systems," in *Intelligent Energy and Power Systems (IEPS), 2014 IEEE International Conference on*, June 2014, pp. 181–185.

[8] M. Reitblatt, M. Canini, A. Guha, and N. Foster, "Fattire: declarative fault tolerance for software-defined networks," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013, pp. 109–114.

[9] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "Fast failure recovery for in-band openflow networks," in *9th International Conference on Design of Reliable Communication Networks, Proceedings*. IEEE, 2013, pp. 52–59.

[10] O. Tilmans and S. Vissicchio, "Igp-as-a-backup for robust sdn networks," in *10th International Conference on Network and Service Management (CNSM)*. IEEE, 2014, pp. 127–135.

[11] T. Pfeiffenberger, J. L. Du, and P. Bittercourt, "Reliable and flexible communications for power systems: Fault-tolerant multicast with sdn/openflow," in *7th IFIP Soft Computing Methods for the Design, Deployment, and Reliability of Networks and Network Applications*, July 2015, pp. 1–6.

[12] P. Pan, G. Swallow, and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels," RFC Editor, RFC 4090, May 2005.

[13] "MPLS Traffic Engineering Fast Reroute – Link Protection - Cisco Systems," URL: http://www.cisco.com/en/US/docs/ios/12_0st/12_0st10/feature/guide/fastrout.html [accessed: 2016-05-17].

[14] M. Pustylnik, M. Zafirovic-Vukotic, and R. Moore, "Performance of the Rapid Spanning Tree Protocol in Ring Network Topology," Siemens AG, Whitepaper, 2007.

[15] "DPDK," URL: http://dpdk.org/ [accessed: 2016-05-17].

[16] "OpenOnload," URL: http://www.openonload.org/ [accessed: 2016-05-17].

 ISBN: 978-1-61208-482-4