Enabling Application Independent Redundancy by Using Software Defined Networking

Armin Veichtlbauer*, Ulrich Pache*, Oliver Langthaler*, Helmut Kapoun†,
Christian Bischof†, Ferdinand von Tüllenburg‡, and Peter Dorfinger‡
* Center for Secure Energy Informatics, Salzburg University of Applied Sciences, Puch/Salzburg, Austria
Email: {armin.veichtlbauer, ulrich.pache, oliver.langthaler}@fh-salzburg.ac.at
† Energy Management, Siemens AG Austria, Vienna, Austria
Email: {helmut.kapoun, bischof.christian}@siemens.com
‡ Advanced Networking Center, Salzburg Research ForschungsgmbH, Salzburg, Austria
Email: {ferdinand.tuellenburg, peter.dorfinger}@salzburgresearch.at

***Abstract*—In critical infrastructures such as power related automation systems (so called “Smart Grids”), reliability is a crucial requirement. In many cases, this is achieved via redundant devices and switch-over functionality. Existing solutions often handle redundancy on the application layer, which can be complex to set up and maintain. This paper presents a novel and generic approach to provide redundancy for Smart Grid systems via software defined networking (SDN) components. Previous approaches of utilizing SDN in Smart Grids have mainly focused on replacing technologies such as Multiprotocol Label Switching (MPLS) and Virtual Local Area Networks (VLANs), in order to reduce configuration effort and to increase network reliability. The proposed concept however aims to simplify the process of setting up control infrastructure redundancy and to increase flexibility. This is achieved by moving the redundancy logic from the application to the network, creating an application agnostic solution to manage redundancy transparently across communication layers. Especially in scenarios where higher layer protocols and policies are tolerant to short interruptions, benefits such as reduced costs and additional configuration possibilities can be expected.**

***Index Terms*—Smart Grid, ICT infrastructure, Redundancy, Virtualization, SDN, IEC 60870-5-104, IEC 61850, Fail-over**

I. Introduction

In automation networks it is common to set up central components in a redundant configuration to avoid single points of failure and thus to guarantee high availability. In many existing power related automation environments, redundancy is provided at component level (using redundant devices) and at application level (e.g., SCADA systems switch over to the backup device after being informed about a failure in the primary system).

The communication infrastructure connecting redundant field devices with highly available backend systems is usually based on Smart Grid typical protocols such as IEC 60870-5-104 (in wide area networks, WANs) and IEC 61850 (in local area networks, LANs) on top of TCP and IPv4. The LAN and WAN areas are usually connected via application layer gateways, which break the continuous TCP connection.

The drawback of this solution is, that the exact hardware architecture as well as the current status of the devices and links has to be known to the SCADA system. In order to separate concerns, i.e., to make the control application independent from the hardware infrastructure, the redundancy of the field infrastructure should be hidden to the SCADA application.

The use case “Virtualized Redundancy” of the funded project “VirtueGrid” aims at setting up a communication environment, which is robust against hardware failures by providing a redundant fail-operational IT subsystem, called the “VirtueGrid communication infrastructure” (VCI). In order to provide the required robustness, other (backup) parts of the VCI have to be able to take over the full functionality in case of failures of the primary system parts.

This leads to the following functional requirements on the VCI for the virtual redundancy use case:

- The solution has to be protocol independent. Consequentely, the solution should be validated with different legacy protocols, such as IEC 60870-5-104, ICE 61850 or DNP3.
- The fail-over time has to be below 5 seconds. A seamless switchover from one gateway to another is preferrable, but not mandatory. A communication breakdown of up to 5 seconds is acceptable for the use case at hand; however, for other use cases, harder timing requirements may apply.
- Fail-over has to be completely transparent to control applications, i.e., independent from application layer functionality. Thus, no redundancy functionality is allowed on application level within the central component (e.g., voters which calculate optimal gateways).

The rest of the paper is structured as follows: Chapter II outlines related scientific work and practical solutions which may have influence to the work at hand. An overview of the hardware and network architecture of the lab prototype, including the test environment for the validation scenarios, is given in chapter III. Chapter IV provides a description of these scenarios, accompanied by the escalation steps that are planned to validate the presented approach. The paper closes with a conclusion section, including proposed further steps in research and development.

II. Related Work

As has been mentioned in the introduction, high availability plays a major role for the operation of electrical systems. Traditionally, this is achieved by introducing redundancy in the electrical and the ICT subsystem. One important challenge is the management of redundancy. Due to the heterogeneity of devices and vendors, standardization and interoperability are hot topics [1], [2]. For management of the redundancy at the ICT subsystem, standards such as virtual local area networks (VLAN) or the more advanced multi-protocol label switching (MPLS) have gained prevalence in current installations. However, these solutions have the drawback that the configuration is error-prone and time consuming.

During the last years, several approaches have been proposed to supplement and replace MPLS and VLANs with software-defined networking (SDN) in Smart Grids, in particular due to its promises to increase flexibility and to simplify network management and due to its network monitoring capabilities [3], [4]. SDN achieves this by basically separating the networks' control plane from its data plane by moving the networks' intelligence to a centralized SDN controller. Using the standardized south-bound-interface, SDN switches (the data plane) communicate to SDN controllers (the control plane) in order to exchange monitoring data and control data. The centralized control plane, for its part, offers a standardized north-bound-interface, providing the possibility for a network owner to tailor network behavior to specific needs [5].

Standardized open interfaces and centralization of control allows for faster implementation of new network functionalities for the Smart Grid than in traditional networks, where new functionalities are based on standards that are required to be implemented (and rolled out) on each particular network device [6]. Examples for fast implementation include the deployment of new routing algorithms implemented at controller level, particularly suitable for power grid communication [7], [8].

Besides this, SDN has been shown to be particularly useful when dependable communication is required like in real-time production machinery networks [9]. Also in context of power grid dependability, SDN has been applied – particularly in three fields related to dependability: Ensuring Quality of Service (QoS), mitigation of cyber-attacks, and mitigation of failures. The heterogeneity of applications in Smart Grids is ranging from energy trading to machine-to-machine (M2M) communication for control of grid devices. All these particular applications open up a diverse and potentially mutually exclusive set of QoS requirements. In general, SDN provides flexible QoS to prioritize critical traffic [10]. With a particular focus on critical traffic flows, for instance switching commands, a behavior similar to DiffServ has been implemented based on SDN [11]. This has been realized by enabling an SDN controller to configure different queues of ports at an SDN Softswitch (Open vSwitch). DiffServ-like, the different queues are reflecting different traffic classes and thus priorities. In evaluation, it has been shown that this approach allows for reliably guaranteed data rates and latencies for a set of selected traffic classes common in power grid environments (IEC 61850 MMS, real-time control, common data transfers). A similar DiffServ inspired approach utilizing queuing capabilities of Open vSwitch, which is controlled by an SDN controller, is applied for enforcing critical IEC 61850 traffic in presence of link flood attacks [12].

Another approach applies the SDN network slicing concept (providing virtually separated network segments for each particular application) for publish-subscribe M2M communication in Smart Grids [13]. Each network slice covers a certain service comprising a group of devices and can be configured with a broad variety of QoS requirements respecting delay or loss. A similar idea was pursued for the development of a software-defined M2M Framework guaranteeing end-to-end QoS by applying network slicing [14]. The slices are generated with respect to QoS requirements of different services. The required information to configure network slices or queues in a suitable manner can be derived from information intrinsic to power grid configurations as has been shown by Molina et al., who used the configuration of electrical substations (encoded in IEC 61850 SCL) to extract logical topology of devices and other relevant information [15]. Also, the benefits of applying QoS network traffic management for power grid frequency control by separating critical traffic classes has been evaluated, showing that the QoS approach led to a "graceful return to the nominal frequency" [16] compared to the reference case and a network load balancing case.

With respect to cyber-attacks, SDN on the one hand opens new threats, but on the other hand opens possibilities for attack mitigation. The opportunities for attack mitigation by utilization of SDN are founded in the SDN capabilities for dynamic (and automated) reconfiguration of the network (e.g., to filter out unwanted traffic), and by building network slices which minimize the effects attacks have on the network [17]. Another case study explains that SDN can be used to hinder eavesdropping of Smart Grid communication by establishing multipath communication [18]. In doing so, each packet of a communication is forwarded along another path through the network under assurance of delivery order. Also proposed has been an encompassing security framework consisting of particular security controllers (besides SDN Controllers) responsible for cryptography, intrusion elimination (IES) and intrusion detection (IDS) [19].

While SDN-based attack mitigation focuses on intentional malicious attacks on the Smart Grid ICT, failure mitigation focuses on unintentional and sudden component failure. Generally, Dorsch et. al state that the failure recovery and resilience of a network and thus the Smart Grid itself can be improved by a "[...] hybrid approach, combining local and centralized methods of failure detection and recovery [...]" [10]. They showed that such an approach allows recovering link failures within the magnitude of milliseconds.

Using wise pre-planning of communication paths in power grid ICT - an approach reminiscent of MPLS fast reroute (MPLS FRR) - has been proposed by Pfeiffenberger et al.

[20]. The approach is based on the idea to establish a fault-tolerant (redundancy implying) multicast tree during network setup. By utilizing OpenFlow's fast-failover groups, OpenFlow switches at junction points of the multicast tree continuously monitor the link states and immediately switch to the backup link as soon as a link failure has been detected. OpenFlow[1] is nowadays the predominant data plane and south-bound interface specification [21].

Another approach for management of redundant communication channels by SDN has been described in form of a demo case by Aydeger et al. [22]. Like in the previous approach, network monitoring capabilities are exploited to immediately switch from a wired communication path to a wireless path as soon as a link failure is detected. However, while Pfeiffenberger et al. exploited OpenFlow's fast failover capabilities for fast switching between redundant paths, Aydeger et al. implemented the switch-over capabilities at controller level, which requires considerably more time for failure detection and link recovery but does not require prior computation of the multicast tree.

Besides the application of SDN to achieve dependable and reliable grid operation, also other ideas exist how to employ SDN to power system communication. For instance, the standard SDN architecture has been adapted to the specific needs of advanced metering by employing bandwidth optimized CoAP [23] to reduce communication overhead and improve control message reliability. Additionally, a more encompassing overview of how SDN has been applied in the area of Smart Grids is given in the survey of Rehmani et al. [24].

Our solution, which we refer to as "virtual redundancy", extends existing research by utilizing SDN in order to achieve transparent redundancy management including seamless integration with legacy power grid components. To the best of our knowledge, there is currently no other approach for applying SDN to provide a means for management of highly available operation of legacy components, which basically relieves legacy components from their duties in this regard.

III. Architecture and Test Environment

To realize the requirements mentioned in chapter I, an SDN approach has been chosen, consisting of an SDN controller and an SDN switch, which forwards data according to the rules given by the controller. In case of failures, the SDN controller will renew the ruleset for the respective switch in order to bypass defective gateways or broken cables. Thus, SDN controllers and SDN switches have been identified as new actors of the ICT infrastructure utilizing an SDN environment. An overview of this architecture is depicted in Fig. 1.

A. Architecture of the ICT Subsystem

Besides the SDN devices, the ICT infrastructure consists of a series of power related devices, which are all connected via IPv4 networks. Hereby, the Main device (C1, see Fig. 1) provides the control logics. In real power environments, a SCADA system could take this role (in this case, the Main device should be realized in a redundant way as well). For our testing purposes, a rack mounted embedded computer is used as Main device.

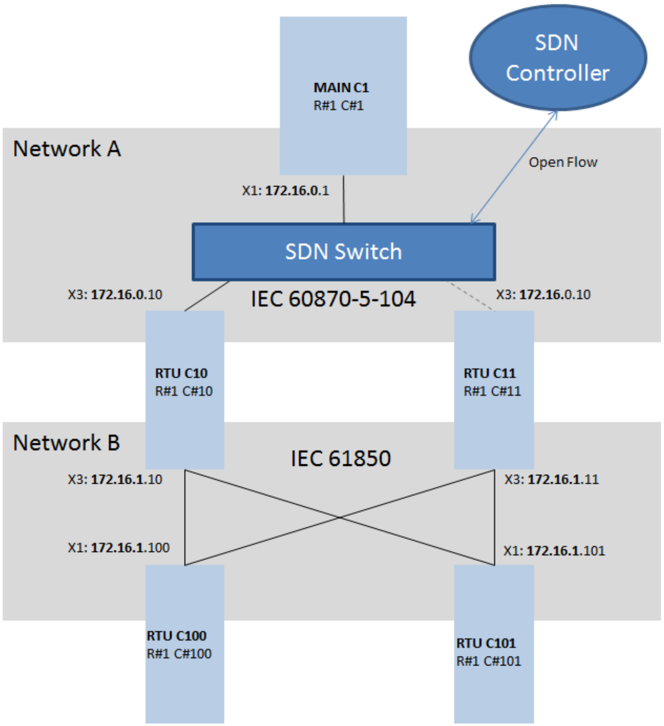

Fig. 1. Network Architecture

As we propose only a redundant ICT infrastructure, power controller redundancy is not in the focus of this work. The integration of a second Main device may cause additional configuration effort at application side, but it is not expected to pose a problem to the ICT infrastructure. Furthermore, the redundant deployment of SDN devices (SDN controllers and SDN switches) would complete the ICT infrastructure; yet, this is considered a solved problem and is thus not further pursued in the test setup at hand.

This Main device is connected to two redundant gateways (C10 and C11) via independent network connections. This WAN part of the VCI is based on a TCP/IP stack, along with IEC 60870-5-104 as application layer protocol. The important point hereby is, that both gateways share the same IPv4 address at the WAN side (yet, they use different addresses at the LAN side), as indicated in Fig. 1.

Finally, at the LAN side (representing the station), we have two "Remote Terminal Units" (RTUs, C100 and C101), which are both connected to both gateways. Thus, a failure of one of the gateways (or of parts of the WAN infrastructure connected to one of the gateways) does not affect the RTUs, provided that at least one gateway is up and running, and that the local connection is working. The LAN side is also based on a TCP/IP stack, but as application layer protocol, IEC 61850 is used. In our test setup, RTUs and gateways are also realized as rack mounted embedded computers, as depicted in Fig. 2.

[1] https://www.opennetworking.org/software-defined-standards/specifications

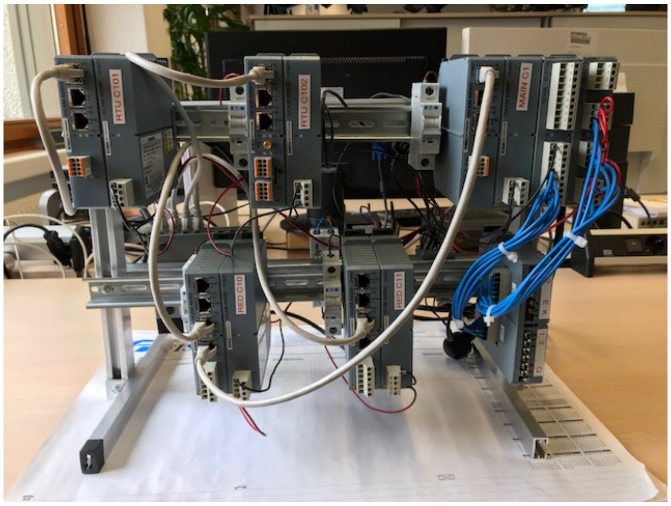

Fig. 2. Laboratory Setup without SDN Switch

B. Setup of Laboratory Environment

As for the scenarios defined in chapter IV, the Main device is used as an emulation of the backend system (e.g., the SCADA system). The scenarios thus are implemented as applications running at this Main device. All other devices may be running on real hardware in laboratory or field environments; or they may be simulated. Thus, a Hardware-in-the-Loop (HiL) setup may be chosen for the tests, as well as a complete simulation setup. In the given case, the hardware shown in Fig. 2 was used.

The application running on the Main device is able to gather information from 6 DIP switches (used to emulate commands, such as to open or to close disconnectors) and to control 6 corresponding LEDs, which indicate e.g. the status of the disconnector or the presence of errors. The application running on C1 does not consider the ICT infrastructure, as conventional control applications would. The only information which is configured, is the IPv4 address of the gateway.

This setup provides a laboratory test environment, which can now be used for conducting a series of tests, intended to provide a proof-of-concept for the SDN based virtualization approach of the ICT infrastructure (VCI).

IV. Scenarios and Test Escalation

In order to generate data for testing this setup, several scenarios have been specified. These scenarios cover control related activities, which are typical for real-life power grids, such as the setting of disconnectors, or general interrogations of field devices data points. Additionally, error-handling scenarios are considered, such as the presence of keep-alive messages or the checking of communication quality by using quality bits. In the following section, the specification of the considered scenarios is described in more detail.

A. Scenarios

1) Disconnection: On the Main device, two DIP switches are evaluated. With the switch S1, the disconnector on RTU 101 can be controlled, whereas with the switch S2, the disconnector on RTU 102 can be controlled. Both simulations are working in the same way.

The state of switch S1 is directly connected to the output of the ON/OFF command. In low state of S1, the OFF command will be sent, in high state, the ON command will be sent (because the input is inverted). The return information (received from RTU 101 and RTU 102) is used to visualize the state of the disconnectors: Lamp 1 represents the state from disconnector 1 and Lamp 2 represents the state from disconnector 2.

The return information is a 2 bit signal and contains the states as shown in Table I:

TABLE I
Return Information and Lamp Settings

	RI OFF	RI ON	Lamp
Disconnector Off	1	0	Off
Disconnector On	0	1	On
Running (no defined state)	0	0	Blinking

To avoid endless blinking in case of a communication error, the quality bits NT (Non-topical) and IV (Invalid) are used to switch the lamp off. The timing of the commands and return information can be seen in Fig. 3.

On the RTU devices, the commands will be received, and the state is stored in a RS Flip Flop. When a new command is received, the return information 0-0 is sent back to the Main device for 5 s. Lastly, the final state is transmitted. Fig. 4 shows the control logics of the disconnector simulation on the Main device.

2) Keep-Alive: For the keep-alive scenario, every second a single command (0.5 s low and 0.5 s high) is sent to the RTUs 101 and 102. If no return information is received from the RTUs for 2 seconds, an error is stored and lamp 6 will be turned on. With switch S6, the error can be acknowledged and reset. If switch S6 is in ON position, the keep-alive application is stopped. In both RTUs (101,102), the received command is sent back directly to the Main device.

3) Communication Error: The protocols IEC 60870-5-104 and IEC 61850 are generating quality bits if an error occurs. This error is shown on lamp 5 (the lamp is switched on via

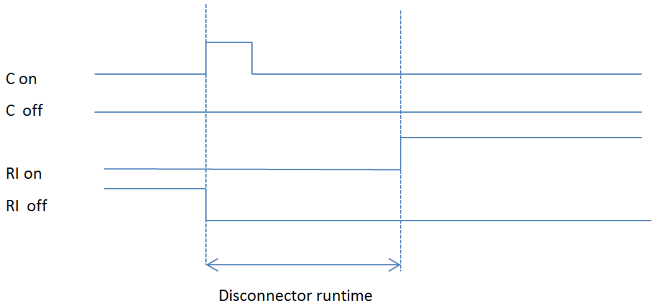

Fig. 3. Disconnector Time Behavior

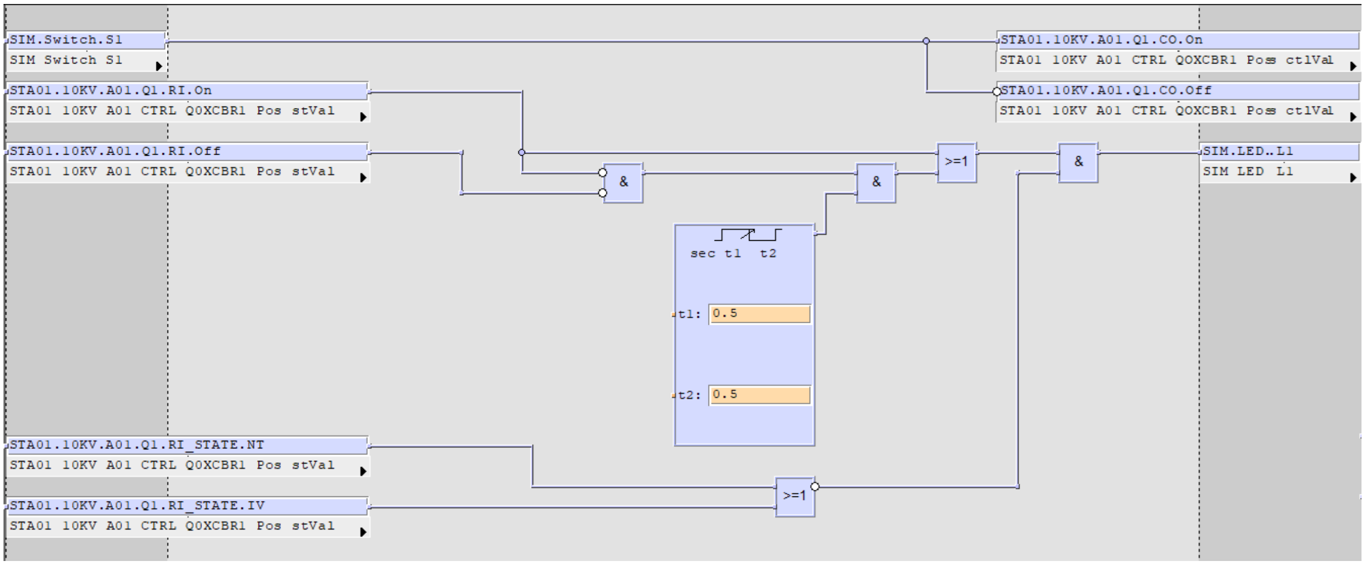
SIM.Switch.S1
SIM Switch S1
STA01.10KV.A01.Q1.RI.On
STA01 10KV A01 CTRL Q0XCBR1 Pos stVal
STA01.10KV.A01.Q1.RI.Off
STA01 10KV A01 CTRL Q0XCBR1 Pos stVal
STA01.10KV.A01.Q1.CO.On
STA01 10KV A01 CTRL Q0XCBR1 Pos ctlVal
STA01.10KV.A01.Q1.CO.Off
STA01 10KV A01 CTRL Q0XCBR1 Pos ctlVal
&
&
>=1
&
SIM.LED..L1
SIM LED L1
sec t1 t2
t1: 0.5
t2: 0.5
STA01.10KV.A01.Q1.RI_STATE.NT
STA01 10KV A01 CTRL Q0XCBR1 Pos stVal
STA01.10KV.A01.Q1.RI_STATE.IV
STA01 10KV A01 CTRL Q0XCBR1 Pos stVal
>=1

Fig. 4. Disconnector Simulation

a simple OR logic). If communication is reestablished, the protocols reset the error bits automatically.

4) General Interrogation (GI): On start-up of an automation unit or individual system elements, or after faults in the system (communication faults, FIFO overflows), the participating automation units or system elements ensure that the operation is resumed automatically in a coordinated manner. This means, that the external and internal communication connections are set up and all data points concerned and relevant system information for the system-wide updating of the process images are again transmitted from their source to their sink. This is done via the initiation of a general interrogation by the station to the corresponding part of the automation network in which the error occurred. In the following cases, a general station interrogation is triggered automatically:

- after power up or reset,
- after communication failure,
- after redundancy switchover to active.

However, the latter case is not applicable to the SDN solution, as it is triggered by the SCADA system (application redundancy). The transmission of “all data concerned” means, that:

- with the station interrogation, all GI-capable messages with process information are transmitted; i.e., all data points that can be interrogated with their state in the periphery and from processing functions (binary information, analogue values, digital values, calculated values, etc.) or system information (system error information),
- in a multi-hierarchical network with a station interrogation not only the local data points of an interrogated automation unit are transmitted, but also those of automation units hierarchically subjacent and reachable over external communication,
- invalid or blocked data points (value disturbances, information affected by the failure of a system element or failure of an automation unit) are also to be transmitted with a station interrogation,
- with a GI request, only those data points are transmitted, which are requested by the station,
- dependent on the process data to be transmitted, the station interrogation is possible both in monitoring as well as in control direction.

B. Test Escalation

For each of these four scenarios, the tests have to show, that the requirements given in chapter I can be met, while keeping the original functionality contained in the chosen scenarios up and running. These tests will be conducted in four stages of expansion, called “test escalation steps”:

- First, the functional scenarios are tested in a conventional communication subsystem environment (without use of SDN) - potentially this may be partly simulated, yet in our setup we use real hardware.
- Second, the functional scenarios are tested in an SDN based communication subsystem (the VCI), but without introducing errors into the VCI.
- Third, the functional scenarios are tested in the VCI with introducing errors (e.g., simulated cable breakage by manually removing cables).
- Fourth, the functional scenarios are tested in the VCI by introducing errors using the SDN configuration itself (this allows for automated testing using appropriate predefined test scenarios).

All scenarios have been implemented by using a function block based IEC 61131 programming environment and deployed to the Main device C1. With these implemented scenarios, the tests relating to the test escalation steps 1 and 2, i.e., without introducing errors to the system, have been conducted. So far, the tests have shown, that without the presence of errors the system is behaving as intended. Hereby, applications are not aware of the underlying ICT infrastructure. The four proposed scenarios are implemented on the basis of existing IEC 60870-5-104 connections to a gateway with a known IPv4 address, without consideration of any network details.

The test escalation steps 3 and 4 are still to be conducted. These tests will show the effects of the SDN fail-over (e.g., fail-over times). However, as the fail-over may take several seconds, this has to be taken into account for applications with hard real-time constraints. Furthermore, the communication is interrupted during the fail-over; for this reason a complete general interrogation has to be carried out (this is also the case if the SCADA system controls switching to a backup device) after the fail-over to ensure that the power controller (SCADA system) receives the current data and the latest status information.

V. Conclusion and Further Work

In this paper, a novel approach to provide redundancy to critical Smart Grid infrastructures has been provided: Redundancy is achieved by using SDN for the ICT infrastructure. As a consequence, the applications (e.g., SCADA systems in the backend) are agnostic to topology changes of the ICT infrastructure of Smart Grids and can therefore work without having to consider the ICT infrastructure.

So far, a laboratory environment has been set up with real components and an emulated backend system. It has already been shown that the network virtualization components, consisting of an SDN switch and an SDN controller, are able to take over all functionalities of the conventional solution, thus finishing the test escalation steps 1 and 2.

Furthermore, this environment can already be used as improved test environment. Hardware failures can easily be simulated by changing the network topology by means of software, which facilitates the scheduling and the implementation of systematic test runs.

The ongoing work will answer the question if, and to which extent, it is possible to run applications in such a test environment in case of errors in parts of the infrastructure. With these applications, several topology changes will be tested, related to test escalation steps 3 and 4. Thereby, transparent fail-over shall be performed. If these tests are successful, further scenarios will be taken into consideration, especially scenarios with harder real-time requirements.

Acknowledgment

The presented work is conducted in the research project VirtueGrid, which is funded by the Austrian Climate and Energy Fund (KLIEN, ref. 858873) within the program e!MISSION (eCall 9096736).

References

[1] M. Emmanuel and R. Rayudu, "Communication technologies for smart grid applications: A survey," *Journal of Network and Computer Applications*, vol. 74, pp. 133–148, Oct. 2016.

[2] J. Kim, F. Filali, and Y.-B. Ko, "Trends and Potentials of the Smart Grid Infrastructure: From ICT Sub-System to SDN-Enabled Smart Grid Architecture," *Applied Sciences*, vol. 5, no. 4, pp. 706–727, Oct. 2015.

[3] S. Rinaldi, P. Ferrari, D. Brandao, and S. Sulis, "Software defined networking applied to the heterogeneous infrastructure of Smart Grid." IEEE, May 2015, pp. 1–4.

[4] Jianchao Zhang, Boon-Chong Seet, Tek-Tjing Lie, and Chuan Heng Foh, "Opportunities for Software-Defined Networking in Smart Grid." IEEE, Dec. 2013, pp. 1–5.

[5] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, Jul. 2013.

[6] A. Cahn, J. Hoyos, M. Hulse, and E. Keller, "Software-defined energy communication networks: From substation automation to future smart grids." IEEE, Oct. 2013, pp. 558–563.

[7] J. Zhao, E. Hammad, A. Farraj, and D. Kundur, "Network-Aware QoS Routing for Smart Grids Using Software Defined Networks," in *Smart City 360*, A. Leon-Garcia, R. Lenort, D. Holman, D. Sta, V. Krutilova, P. Wicher, D. Cagov, D. pirkov, J. Golej, and K. Nguyen, Eds. Cham: Springer International Publishing, 2016, vol. 166, pp. 384–394.

[8] M. Alishahi, M. H. Yaghmaee Moghaddam, and H. R. Pourreza, "Multi-class routing protocol using virtualization and SDN-enabled architecture for smart grid," *Peer-to-Peer Networking and Applications*, vol. 11, no. 3, pp. 380–396, May 2018.

[9] M. Herlich, J. L. Du, F. Schorghofer, and P. Dorfinger, "Proof-of-concept for a software-defined real-time ethernet." in *ETFA*, 2016, pp. 1–4.

[10] N. Dorsch, F. Kurtz, S. Dalhues, L. Robitzky, U. Hager, and C. Wietfeld, "Intertwined: Software-defined communication networks for multi-agent system-based Smart Grid control." IEEE, Nov. 2016, pp. 254–259.

[11] N. Dorsch, F. Kurtz, H. Georg, C. Hagerling, and C. Wietfeld, "Software-defined networking for Smart Grid communications: Applications, challenges and advantages." IEEE, Nov. 2014, pp. 422–427.

[12] H. Maziku and S. Shetty, "Software Defined Networking enabled resilience for IEC 61850-based substation communication systems." IEEE, Jan. 2017, pp. 690–694.

[13] Y.-J. Kim, K. He, M. Thottan, and J. G. Deshpande, "Virtualized and self-configurable utility communications enabled by software-defined networks." IEEE, Nov. 2014, pp. 416–421.

[14] Z. Zhou, J. Gong, Y. He, and Y. Zhang, "Software Defined Machine-to-Machine Communication for Smart Energy Management," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 52–60, Oct. 2017.

[15] E. Molina, E. Jacob, J. Matias, N. Moreira, and A. Astarloa, "Using Software Defined Networking to manage and control IEC 61850-based systems," *Computers & Electrical Engineering*, vol. 43, pp. 142–154, Apr. 2015.

[16] A. Sydney, D. S. Ochs, C. Scoglio, D. Gruenbacher, and R. Miller, "Using geni for experimental evaluation of software defined networking in smart grids," *Computer Networks*, vol. 63, pp. 5–16, 2014.

[17] X. Dong, H. Lin, R. Tan, R. K. Iyer, and Z. Kalbarczyk, "Software-Defined Networking for Smart Grid Resilience: Opportunities and Challenges." ACM Press, 2015, pp. 61–68.

[18] E. Germano da Silva, L. A. Dias Knob, J. A. Wickboldt, L. P. Gaspary, L. Z. Granville, and A. Schaeffer-Filho, "Capitalizing on SDN-based SCADA systems: An anti-eavesdropping case-study." IEEE, May 2015, pp. 165–173.

[19] U. Ghosh, P. Chatterjee, and S. Shetty, "A Security Framework for SDN-Enabled Smart Power Grids." IEEE, Jun. 2017, pp. 113–118.

[20] T. Pfeiffenberger, J. L. Du, P. B. Arruda, and A. Anzaloni, "Reliable and flexible communications for power systems: Fault-tolerant multicast with SDN/OpenFlow." IEEE, Jul. 2015, pp. 1–6.

[21] F. Hu, Q. Hao, and K. Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 24.

[22] A. Aydeger, K. Akkaya, and A. S. Uluagac, "SDN-based resilience for smart grid communications." IEEE, Nov. 2015, pp. 31–33.

[23] J. Kim, F. Filali, and Y.-B. Ko, "A lightweight CoAP-based software defined networking for resource constrained AMI devices." IEEE, Nov. 2015, pp. 719–724.

[24] M. H. Rehmani, A. Davy, B. Jennings, and C. Assi, "Software Defined Networks based Smart Grid Communication: A Comprehensive Survey," *arXiv:1801.04613 [cs]*, Jan. 2018.