

# Enabling Hard Services Guarantees in Software-Defined Smart Grid Infrastructures

Nils Dorsch\*, Fabian Kurtz, Christian Wietfeld

*TU Dortmund University  
Otto-Hahn-Str. 6, 44227 Dortmund, Germany*

---

## Abstract

Information and Communication Technology (ICT) infrastructures play a key role in the evolution from traditional power systems to Smart Grids. Increasingly fluctuating power flows, sparked by the transition towards sustainable energy generation, become a major issue for power grid stability. To deal with this challenge, future Smart Grids require precise monitoring and control, which in turn demand for reliable, real-time capable and cost-efficient communications. For this purpose, we propose applying Software-Defined Networking (SDN) to handle the manifold requirements of Smart Grid communications. To achieve reliability, our approach encompasses fast recovery after failures in the communication network and dynamic service-aware network (re-)configuration. Network Calculus (NC) logic is embedded into our SDN controller for meeting latency requirements imposed by the standard IEC 61850 of the International Electrotechnical Commission (IEC). Thus, routing provides delay-optimal paths under consideration of existing cross traffic. Also, continuous latency bound compliance is ensured by combining NC delay supervision with means of flexible reconfiguration. For evaluation we consider the well-known Nordic 32 test system, on which we map a corresponding communication network in both experiment and emulation. The described functionalities are validated, employing realistic IEC 61850 transmissions and distributed control traffic. Our results

---

\*Corresponding author

*Email address:* [nils.dorsch@tu-dortmund.de](mailto:nils.dorsch@tu-dortmund.de) (Nils Dorsch)

show that hard service guarantees can be ensured with the help of the proposed SDN solution. We identify extremely time critical services, which must not be subjected to flexible reconfiguration.

*Keywords:* Smart Grid Communications, Mission Critical Systems, Hard Service Guarantees, Software-Defined Networking, Network Calculus.

---

## 1. Introduction

Future power systems are faced with severe challenges, caused by the transition from conventional to distributed, renewable generation [1]. To fully exploit the advantages and mitigate the drawbacks of fluctuating power generation from these energy resources, concepts such as Demand Side Management (DSM) and controllable loads/storages, e.g. scheduling Electric Vehicle (EV) charging, need to be applied. At the same time, the energy system has to deal with further volatile power transmissions, caused by increasing energy trade due to the liberalization of energy markets. Resulting from these challenges, precise monitoring and control of the system are indispensable for maintaining grid stability and avoiding cascading outages. Subsequently, appropriate Information and Communication Technology (ICT) infrastructures are required to ensure reliable, timely transfer of measurement data and control commands, in particular on transmission grid level [2, 3]. The International Electrotechnical Commission (IEC) defines in its standard IEC 61850, which is set to become the prevailing normative for power grid communications, intervals as low as  $250 \mu\text{s}$  and maximum allowed latencies of 5 ms for measurement data transmission respectively protection tripping [4]. Meanwhile, distribution grid communications deal with numerous protocols and a variety of different access technologies [5]. Overall, an increasing number of Intelligent Electronic Device (IED), each with distinct service requirements, will be connected to wide area communication networks.

To cope with these specific demands of Smart Grid communications, we propose a comprehensive framework, building on the concept of Software-Defined

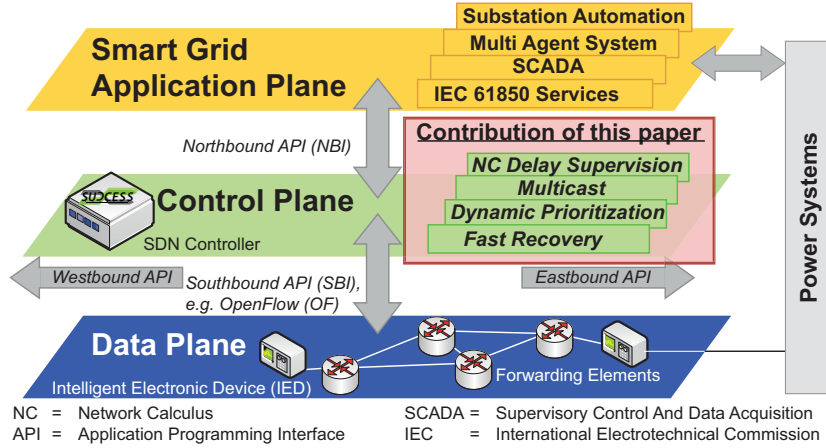


Figure 1: Solution approaches addressed in this paper, mapped on the Software-Defined Networking for Smart Grids concept, introduced in [9]

25 Networking (SDN). In this way, we are able to provide hard service guarantees with traffic flow granularity. SDN constitutes a promising new take on network-  
 ing, offering flexible, dynamic configuration of communication infrastructures [6]. Following the paradigm of separating data and control planes, SDN estab-  
 30 lishes a programmable controller platform, with global network view, capable of managing traffic flows. While various mechanisms for enhancing particular aspects of communications' Quality-of-Service (QoS) exist, they typically suffer from vendor specific peculiarities, poor integration and overly complicated configuration [7]. In contrast, our approach is able to address the multitude of  
 35 diverging requirements, while allowing for straight-forward extension and configuration. In particular, this concept provides means for fast failure recovery, dynamic prioritization and queue configuration under the overall paradigm of application- and QoS-awareness. Network Calculus (NC) algorithms [8] are incorporated into our SDN controller to predict and monitor end-to-end delays of traffic flows analytically. Hence, violations of delay bounds can be identified  
 40 in time to activate counter-measures, ensuring continuous fulfillment of hard real-time guarantees.

The main contributions of this paper are the following:

- Software-Defined Networking enabled service-centric network configuration and adaption for Smart Grids, providing hard service guarantees
- 45 • the integration of NC into SDN-driven network control for delay supervision and routing to ensure real-time capable communications at all times

Figure 1 provides an overview of our concept for SDN-enabled Smart Grid communications, highlighting interactions between ICT and power system applications. We evaluate our concepts, considering IEC 61850 communications as well as a Multi-Agent System (MAS) for distributed control on a fiber-based communication infrastructure for the Nordic 32 test system [10]. Both empirical measurements - on a subset of the whole test system - and emulations of the whole infrastructure are utilized. In addition, the proposed concepts may be adapted to other mission critical systems such as transportation or rescue services.

This work has been carried out as part of larger scale research efforts, i.e. DFG research unit 1511 and the Franco-German project BERCOM. Subsequently, Smart Grid requirements were synchronized and solution approaches discussed with power system experts and utilities such as EDF. The remainder of this work is structured as follows: Section 2 provides an overview of the state-of-the-art, detailing the requirements of Smart Grid communications and introducing the main principles of SDN and NC. The section is completed by an overview of related work. Next, we describe our solution approach based on the SDN controller framework (Section 3). In Sections 4 and 5 a description of the Smart Grid scenario and an overview of the developed testbed setup are provided. Afterwards, empirical, emulation and analytical evaluation results are presented in Section 6. Finally, the paper concludes with a summary and an outlook on future work (Section 7).

## 2. State-of-the-Art on Smart Grid Communications, Software-Defined Networking and Associated Performance Evaluation

This section reviews Smart Grid communication requirements and reflects on the state-of-the-art of Software-Defined Networking (SDN) and Network Calculus, for enabling respectively verifying hard service guarantee compliance. Afterwards results of related work are described and compared to this article.

### 2.1. Smart Grid Communication Use Cases

Smart Grid communication requirements can be roughly divided into distribution and transmission grid use cases, as detailed below. While these power system levels exhibit widely diverging demands, SDN offers integrated approach for associated communications.

#### 2.1.1. Managing the Distribution Power Grid

Communication-dependent applications in the distribution power grid comprise Automated Meter Reading (AMR), DSM, monitoring and control of Distributed Energy Resources (DER) as well as coordination of EV charging. Design and operation of ICT infrastructures are driven by large numbers of devices, heterogeneity of protocols and technologies [11]. While IEC 61850 becomes increasingly important for DER control, dedicated sets of protocols are applied for AMR (e.g. IEC 62056, DLMS/COSEM) and EV charging (e.g. ISO 15118 and OCCP). For physical transmission, various wired (Powerline Communications (PLC), broadband cable) and wireless access technologies (WiFi, cellular) are considered. Moreover, driven by business-to-consumer use cases, aspects like role management, authentication and billing play an important role.

#### 2.1.2. Controlling the Transmission Power Grid

In contrast to distribution systems, communications on the transmission grid level focus on requirements such as reliability, real-time capability and security. Use cases involve substation automation including extremely time critical protection functions, Wide-Area Monitoring Protection and Control (WAMPAC)

Table 1: Smart Grid timing requirements, specified in IEC 61850-5 [4]

<b>Transfer Time Class</b>	<b>Maximum Transfer Time [ms]</b>	<b>Type of Transfer</b>
0	> 1000	files, events, logs
1	1000	events, alarms
2	500	operator commands
3	100	slow automatic interactions
4	20	fast automatic interactions
5	10	releases, status changes
6	3	trips, blockings

and Supervisory Control and Data Acquisition (SCADA). Fiber-optic infrastructures are regarded as main transmission medium, whereas cellular networks are considered as alternative or back-up solution for the network access domain.

100

*Centralized Power System Control.* SCADA provides the basis for centralized grid control functionalities. Protocol-wise IEC 60870 is currently still widely applied for this purpose. However, IEC 61850, originating from substation automation, is about to become the dominating protocol throughout transmission system communications (as well as for some distribution grid applica-  
 105 tions). It employs a holistic approach, covering detailed data models for devices and functions, abstract communication service descriptions as well as actual protocols. Measurement values are transmitted in fixed intervals of  $250 \mu\text{s}$ , using Sampled Value (SV) messaging. The Generic Object Oriented Substation  
 110 Event (GOOSE) service is applied for exchanging statuses and issuing switching commands. Both message types are encapsulated into Ethernet packets directly. GOOSE operates semi-regularly with periodic status messages in intervals of e.g. 1 s, whereas commands are issued in response to events and are repeated in increasing intervals starting at 1 ms. Meanwhile, Manufacturing  
 115 Message Specification (MMS) utilizes client-server-based TCP/IP communica-

tion for tasks like software updates, configuration and measurement reports. Table 1 provides an overview of end-to-end timing demands for different applications in IEC 61850, regardless of communication failures. The requirements are divided into corresponding Transmission Traffic Classes (TTC), defining  
120 maximum transfer times.

*Distributed Power System Control.* Differing from the common SCADA approach, power systems may also be controlled in a distributed manner, utilizing for example a MAS. Such an MAS is introduced in [12], placing agents at substations of the power grid. These agents utilize local information along with  
125 data from adjacent substations, received via inter-agent communication, to gain an estimate of the surrounding power grid's state. In case emergency conditions are detected, the agents coordinate counter-measures and apply local assets to stabilize voltage and prevent black-outs. For example, set points of High Voltage Direct Current (HVDC)-converters and power flow controllers can be changed or  
130 re-dispatch of flexible generation and load may be initiated. A first integration between a JAVA-based implementation of this distributed grid control and our SDN controller framework was achieved in [13].

## 2.2. Software-Defined Networking Enabled Communication Systems

Software-Defined Networking is a novel approach towards networking, based  
135 on the idea of separating control and data plane [6]. Therefore, control functionalities are abstracted from switches or routers and consolidated at a dedicated control instance, known as the SDN controller. Subsequently, switches and routers on the data plane, i.e. forwarding elements, handle physical transmission of packets only, forwarding unknown traffic flows to the SDN controller.  
140 This central component handles routing and installs corresponding forwarding rules at all relevant devices throughout the network. Subsequent packets of the same traffic flow are handled by the data plane components on basis of the rules established previously. Communication between the SDN controller and the forwarding elements is handled via the so-called Southbound Interface (SBI) with

145 Open Flow (OF) [14] being the most prominent - de-facto standard - protocol  
for this purpose [15]. One major benefit of SDN is the controller's programmability, which - in conjunction with its global network view - can be used to adapt dynamically to changes in the communication network. Moreover, it allows for straightforward integration of a variety of different approaches and algorithms,  
150 like for example traffic engineering capabilities of Multi Protocol Label Switching (MPLS). While integrating such functionalities, SDN obviates overly complex configuration, usually associated with such approaches [7]. Thus, network management and control are simplified significantly. Through its Northbound Interface (NBI) the SDN controller discloses means of conveying communication requirements and influencing network behavior to external applications.  
155 Contrary to the SBI, there is no common protocol for the NBI, though the Representational State Transfer (REST) Application Programming Interface (API) is in widespread use [16]. To achieve scalability of the SDN approach, i.e. for controlling large infrastructures, interaction with other controllers and legacy  
160 networks is enabled via the westbound and eastbound interface respectively.

Today, SDN is already widely deployed in data centers of companies such as Alphabet/Google [17] and is considered as the foundation for wired communications in 5G mobile communication networks [18].

### 2.3. *Network Calculus for the Performance Evaluation of 165 Communication Infrastructures*

To obtain a precise, real-time view on the delay of Smart Grid communications, NC is integrated into the controller framework as an analytical modeling approach for delay computation. NC, originating from the initial works of Cruz [19] in the early 1990s, is a well-established method for the worst-case analysis  
170 of communication networks. It is suited for arbitrary types of traffic as the approach is agnostic to statistical distribution functions, providing performance bounds only. Current advancements of NC favor the use of tighter, stochastic bounds, which come at the price of small violation probabilities. In this work, however, the original, deterministic NC is applied, as timing requirements of



175 communications in transmission power grids are extremely strict and violations may result in a fatal collapse of the system. Hence, thorough, deterministic delay bounds, excluding any violations, are considered most suitable.

Originating from NC terminology, we introduce *flow-of-interest* and *cross traffic flows* as major terms for describing network behavior in this article.

- 180
- **Flow-of-interest** refers to the packet transfer, which is in the current focus of analysis.
  - **Cross traffic flows** are other transmissions, that are concurrently active on the same network and may interfere with the flow-of-interest.

To model traffic, arriving to the communication system, we employ the frequently used, leaky (token) bucket arrival curve in Equation 1.

$$\alpha(t) = \sigma + \rho \cdot t, \quad (1)$$

where  $\sigma$  is the maximum packet size and  $\rho$  the sustained data rate requirement of the traffic flow. These parameters follow pre-defined values per assigned traffic/priority class. To map the service, which is offered to the traffic flow by network elements such as links or switches, the concept of service curves is adopted. Here, we use rate latency curves per outgoing switch port, considering data rate  $R$  and propagation delay  $T_{pr}$  of the link as well as transmission ( $T_{tr}$ ) and switching delay ( $T_{sw}$ ):

$$\beta(t) = R \cdot [t - T]^+, \quad (2)$$

with  $T = T_{pr} + T_{tr} + T_{sw}$ . By linking arrival and service curves, the delay and backlog, that is experienced by the flow-of-interest at the respective network element, can be determined. To obtain the traffic flow's overall network delay bound directly, NC utilizes the concept of the end-to-end service curve. It is calculated as the convolution of all service curves on the flow's path, as given

by Equation 3.

$$\beta_{end-to-end,i}(t) = \beta_{1,i}(t) \otimes \dots \otimes \beta_{n,i}(t), \quad (3)$$

with  $1..n$  being the index of the switches on the path between source and destination. The interference of other transmissions, *cross-traffic flows*, is captured by the left-over-service curve  $\beta_{k,i}(t)$  with  $i$  being the index of the flow-of-interest and  $k$  identifying the respective switch. It is defined by Equation 4 and describes the service, which can still be provided to the flow-of-interest after taking into account interfering traffic.

$$\beta_{k,i}(t) = \beta_{k,base_i}(t) - \sum_{j=i}^m (\alpha_{k,j}(t - \Theta)), \quad (4)$$

where cross traffic flows of same or higher priority ( $j = i..m$ ) reduce the service available to flow  $i$ . Subsequently, the cross traffic arrival curves  $\alpha_{k,j}$  of flow  $j$  at node  $k$  are subtracted from the specific base service curve of flow  $i$ . For flows of higher priority ( $j > i$ ) strict prioritization is assumed, resulting in  $\Theta = 0$ , whereas for flows of the same priority First In First Out (FIFO) scheduling applies, introducing  $\Theta$  as additional level of flexibility.

#### 190 2.4. Related Work

In recent years, SDN has been a major topic of research with numerous related publications. Hence, our review focuses on a subset of these works, i.e. papers which apply SDN in the context of Smart Grids or aim at integrating SDN with NC.

195 Starting with the latter, Guck *et al.* split online routing and NC-based resource allocation, achieving average link utilization close to the results of mixed-integer programming in software-defined industrial ICT infrastructures [20]. In contrast to our approach, performance is assessed individually for each node, instead of applying end-to-end bounds, which are known to be tighter. NC is applied in [21] to create a high-level abstraction model of network service ca-  
200

pabilities, guaranteeing inter-domain end-to-end QoS. Thus, the authors derive the required bandwidth of services, whereas this work focuses on end-to-end latency guarantees. In [22] a variation of NC serves as basis for a multi-constraint flow scheduling algorithm in SDN-enabled Internet-of-Things infrastructures. The performance of SDN deployments is evaluated, modeling SDN controller-switch interactions with NC in [23]. Yet, computations are performed offline as the approach is not coupled with an actual SDN set-up. Similarly, Huang *et al.* validate their proposed hybrid scheduling approach for SDN switches by applying offline NC analysis [24]. In [25] NC is employed for the analysis of SDN scalability. Therefore, the authors determine worst case delay bounds on the interaction between network nodes and SDN controller. The approach considers switch internals and utilizes similarities between flow tables and caches. Evaluations indicate sensitivity to parameters such as network and flow table size, traffic characteristics and delay, allowing to deduce recommendations for distributed controller concepts. Just as the previous two articles, this publication analyzes SDN-enabled infrastructures with the help of NC, but does not integrate it with the system. In previous studies we modeled a traditional wide-area communication network for transmission systems on basis of IEC 61850 and evaluated its real-time capability using NC [26]. The developed framework serves as a starting point for combining NC and SDN within this article.

A general overview of possible applications of SDN in Cyber-Physical Systems (CPSs) is given in [27]. With regard to Smart Grid communications, Cahn *et al.* proposed SDN-based configuration of a complete IEC 61850 substation environment [28]. Molina *et al.* propose an OF-enabled substation infrastructure, integrating IEC 61850 configuration into the Floodlight controller by reading Substation Configuration Description (SCD) files [29]. In this way, the approach is very similar to the concepts presented in [28]. Based on the configuration file, static traffic flows with different priorities are established. Mininet is employed to test functionalities such as traffic prioritization, detection of Denial-of-Service (DoS) attacks and load balancing. However, these use cases show only minor advancements compared to standard Floodlight, whereas

the main contribution is automatic substation network configuration. In [30] SDN is utilized to design a network intrusion detection system for SCADA communications. To facilitate the communication between smart meters and  
235 the control centers, aggregation points are introduced to the SDN data plane in [31]. Planning of these is optimized with respect to minimal costs applying a mathematical model. In [32] SDN is used for establishing networked microgrids, enabling event-triggered communication. The authors claim to reduce costs and enhance system resilience in this way. All of the former publications illustrate  
240 specific applications of SDN in Smart Grids not within the scope of this article. Meanwhile, these papers do not study the particulars of hard service guarantees addressed here.

Sydney *et al.* compare MPLS- and OF-based network control for power system communications, demonstrating that SDN achieves similar performance,  
245 while simplifying configuration [7]. The author expanded their work by experiments on the GENI testbed [33]. Evaluations are performed using the example of demand response, where load shedding is triggered to maintain frequency stability. In this context, three functionalities are tested: fast failover, load balancing and QoS provisioning. Thus, the paper addresses topics quite similar  
250 to this article. However, no standard Smart Grid communication protocol is applied. Also, the publication is rather focused on the electrical side, whereas some communication aspects are not studied in full detail. For example, the presented recovery process is comparably slow with delays of up to 2s and would require further optimization. In addition, our investigation considers further  
255 functionalities such as dynamic network reconfiguration and delay supervision. Mininet emulation, integrated with ns-3 simulation, is used in [34] to evaluate SDN-based failure recovery to wireless back-up links in a Smart Grid scenario. OF Fast Failover Groups (FFG) are used in [35] to enable fault-tolerant multi-cast in Smart Grid ICT infrastructures.

260 In previous work we proposed an SDN controller framework, which provides fault tolerance and dynamically adaptable service guarantees for Smart Grid communications [9, 36, 13]. Compared to these publications and other related

work discussed above, we achieve the following improvements and contributions in this work:

- 265 • comprehensive comparison of different fast recovery approaches, quantifying path optimality and detection overhead in addition to recovery delays
- delay impact of dynamic network reconfiguration in response to Smart Grid service requirements and network conditions, illustrated on a five step sequence of events
- 270 • delay-aware routing using NC
- compliance to hard service guarantees on basis of NC delay supervision

### 3. Proposed Solution Approach for Smart Grid Communications on Basis of Software-Defined Networking

To address the challenges of communications in critical infrastructures such as the Smart Grid, we propose the Software-Defined Universal Controller for Communications in Essential Systems (SUCCESS). It is a Java-based framework, designed to meet hard service requirements of mission critical infrastructures.

Even though in the following we refer to the control plane as a single instance, we acknowledge the need for deploying distributed or hierarchical systems of multiple controllers for large-scale real world scenarios. Otherwise, real-time reconfiguration of such infrastructures could not be achieved. Yet, inter-controller coordination is not within the scope of this work.

Figure 2 illustrates the different components of our controller, including their interdependencies as well as the connection to Smart Grid applications via the Northbound Interface (NBI). As a basis for the main contributions of this work, we devise the following functions:

- *Global Network State Monitoring*: Active traffic flows as well as link states are tracked to obtain a real-time view of the current network load.

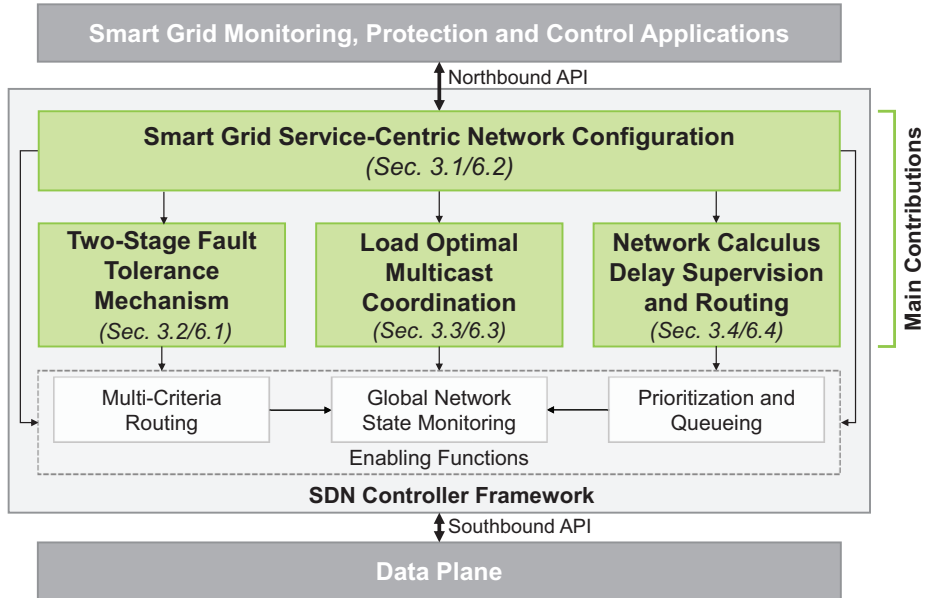


Figure 2: Elements of the Software-Defined Universal Controller for Communications in Essential Systems, their interdependencies and classification within the SDN concept (including reference to corresponding discussions)

- 290 • *Multi-Criteria Routing*: In contrast to standard optimal path routing, we employ Depth-First Search (DFS) to determine multiple feasible routes, which can be applied as alternatives for fast failure recovery and hard service guarantee provisioning.
- 295 • *Prioritization and Queueing*: For prioritization we apply large range of priority levels, which are mapped to corresponding queues, which encompass minimum and maximum data rate guarantees on basis of Linux Hierarchical Token Bucket (HTB) [37]. We enable controller-driven, flexible queue configuration using Open vSwitch (OVS) Database (DB) commands via Secure Shell (SSH).

### 300 3.1. Smart Grid Service-Centric Network Configuration

For adapting communication network configurations to Smart Grid specific requirements, we enable power system applications to convey their demands to the controller. Therefore, we implement the SDN NBI, using the REST API.

While the controller is set up as the REST server, applications act as clients,  
305 sending requests to the controller. Interaction via the NBI is demonstrated  
employing the MAS as client application. Four different services - Rule Cre-  
ation, Route Reservation, Flow Modification, Multicast Group Creation and  
their respective revocations - are provided by the controller. Details on these  
NBI services are provided below.

#### 310 *3.1.1. Rule Creation*

Rule Creation serves to register traffic flows at the controller, disclosing their  
specific demands regarding minimum data rate, maximum latency and packet  
loss as well as priority. This information is stored at the controller as combined  
flow requirements. Thus, incoming traffic can be routed and directed to an  
315 adequate priority queue, fulfilling its requirements. Hence, this functionality  
relies heavily on the routing, prioritization and queuing mechanisms, described  
previously. Applying the DELETE command in conjunction with Rule Creation  
removes the respective traffic rule.

#### *3.1.2. Route Reservation*

320 Typically, in SDN-enabled infrastructures network devices request routes  
for new packet streams from the controller. This incurs additional delay for the  
first packets of a transmission. Route Reservation, however, is applied to route  
traffic flows and configure flow table entries in advance, avoiding this initial  
delay. However, such static flow table entries need to be removed explicitly,  
325 since idle time-outs are precluded.

#### *3.1.3. Flow Modification*

Existing flow requirements, involving priority and queue assignments, may  
be altered using this request. Hence, it becomes possible to raise or reduce  
flow priorities temporarily, e.g. in response to emergency situations. In partic-  
330 ular, this request may be performed in case of simultaneous overloads of power  
and communication system. Thus, successful transmission of critical commands  
for relieving the power grid crisis can be ensured. Temporary changes to the

flow requirements can be revoked with the help of the corresponding DELETE command.

#### 335 3.1.4. Multicast Group Creation

We provide dedicated NBI requests, enabling Smart Grid applications to trigger generation, modification and deletion of multicast groups. To create a new multicast group, the controller is supplied with a list of Media Access Control (MAC) or Internet Protocol (IP) addresses, representing member de-  
340 vices. In addition, a set of header defines the messages, applicable for multicast transmission. Hence, the controller is able to identify multicast packets and determine appropriate routes to all destinations. The use of specific multicast addresses is not required.

#### 3.1.5. Further Aspects of Smart Grid Adaptation

345 Besides the aforementioned means of direct participation, SDN provides further benefits, facilitating Smart Grid communications. As IEC 61850 is becoming a comprehensive standard for power systems, its application for wide area communications is discussed. Technical reports propose the transmission of Ethernet-based SV and GOOSE messages over IP systems, necessitating tun-  
350 neling or conversion of packets to routableGOOSE/routableSV [38]. In contrast, packet routing and forwarding in OF-enabled infrastructures builds on matches - sets of arbitrary header fields - and thus is protocol-agnostic. This allows for direct transmission of IEC 61850 SV and GOOSE messages on wide area networks.

#### 355 3.2. Two-Stage Fault Tolerance Mechanism

Guaranteeing reliable, virtually uninterrupted, transmission is a major requirement for mission-critical communications. Therefore, mechanisms enabling fast recovery after link failures are integrated into the controller. Failover can be split into two steps: failure detection and traffic restoration. Both can be  
360 realized either locally at the switches or centrally, triggered by the SDN con-



troller. Building on the advantages of central and local algorithms, a hybrid two-stage solution is proposed.

*BFD-based Local Recovery.* Bidirectional Forwarding Detection (BFD) [39] is deployed to reduce failure detection times locally at the switches. It is integrated into OVS since version 2.3.0 [40] and is also applied in combination with MPLS Fast Reroute (FRR) to achieve fast recovery in MPLS-based infrastructures [41]. For monitoring a link, BFD sends lightweight messages in fixed intervals between two switches, connected via a link. If no packets from the other end of the communication line are received within a defined multiple of the packet Inter-Transmission Time (ITT), the link is assumed to have failed. Here, the ITT may be as low as 1 ms, while the usual detect multiplier amounts to 3. Reaction to link failures, detected by BFD, can be realized locally using OF Fast Failover Groups (FFG). Therefore, after completing routing of a traffic flow, the controller determines alternative switch configurations for every possible link failure within the main path. These alternative configurations are stored in the switches' forwarding tables along with the main path using FFG. Thus, in case the outgoing port of a traffic flow is reported as failed, the flow is switched to its alternative path automatically. To reduce the number of additional forwarding table entries at the switches, our algorithm is designed to maximize the similarity between main and recovery path, letting the traffic flow return to its initial path after as few hops as possible.

*SDN-driven Central Recovery.* For centralized link status monitoring, we devise a heartbeat mechanism, similar to BFD, which regularly transmits lightweight probe packets. However, in this case packets are sent out by the controller, thus consuming bandwidth of control and data network. Encapsulated into OFPacketOut messages, heartbeat packets are transferred to the switches, which extract and forward the content on the monitored link. At the other end of the link, the packet is sent back to the controller using the OFPacketIn format. If this packet is not returned to the controller within a defined interval, the link is classified as failed. In contrast to local failover, recovery paths are not

pre-computed, but determined on-demand, considering current network load for obtaining load/latency optimal routes.

*Two-Stage Hybrid Recovery.* Local failover mechanisms usually achieve faster traffic recovery compared to centralized approaches, yet might employ sub-optimal paths, resulting in network overload. Vice versa, controller-driven recovery enables optimal traffic configuration at all times, while failover times are considerably higher. Subsequently, a hybrid approach, combining the advantages of local and central mechanisms, is created. First, BFD is employed for detecting link failures and traffic is immediately switched to intact paths with the help of FFG. Next, the controller is notified of the failure, new optimal paths are determined and corresponding forwarding table entries are installed at the switches. Thus, fast recovery is realized, while time intervals of sub-optimal traffic flow respectively network configuration are minimized.

### 3.3. Load Optimal Smart Grid Multicast Coordination

Applying multicast flows allows for significant network load reductions. This is achieved by utilizing a shared path for packets from one source to multiple destinations for as long as prudent. While this concept is well-known in conventional communication networks, it is applied infrequently due to the significant effort associated with the configuration and management of multicast groups. However, this technique plays an important role in IEC 61850-based communication, being applied for the distribution of measurement values and status updates. In this work, setup and maintenance of multicast groups is facilitated by providing direct access via the SDN NBI, as detailed in Section 3.1. The Smart Grid application simply has to provide a list of intended group members in terms of IP or MAC addresses along with a set of packet matching criteria. After reception of the first packet, which matches the multicast group, the controller performs routing and forwarding rule setup. To enable multicast handling, paths are defined as routing trees. For routing, we implemented the Bounded Shortest Multicast Algorithm (BSMA) [42], which minimizes the

420 number of used links, while at the same time fulfilling flow requirements such  
as maximum delay bounds.

#### 3.4. Network Calculus-Based Delay Supervision and Routing

Other than in legacy networks, where NC can be applied for offline performance evaluation only, SDN allows for utilizing this analytical technique during  
425 live operation. For this purpose, we integrate NC logic into the SDN controller to achieve - guaranteed - compliance to defined real-time requirements of Smart Grids at all times. A corresponding overview of latency demands is given in Table 1 with requirements ranging from 3 ms to more than 1 s. To pursue the goal of real-time capable communications, NC is applied for the following two  
430 use cases:

- **routing of new traffic flows:** provide delay-optimal paths, complying to given latency requirements
- **monitoring of existing traffic flows:** ensure delay bound compliance, even when (other) flows are reconfigured or new flows are added

435 Before going into the details of these tasks, necessary extensions and modifications of NC are described in the following section.

#### 3.5. Queue Rate and Cross Traffic Extensions to Network Calculus

Complex Smart Grid infrastructures and diverse traffic flows require a detailed study of cross traffic impact as they may lead to non-feed forward behavior, which continues to be an issue of NC analysis. In addition, the influence of  
440 HTB scheduling has to be considered in NC evaluations.

Beginning with the latter aspect, we enhance our NC framework to consider minimum and maximum queue rates as introduced in Subsection ??, thus reflecting the preconditions of our testing environment. Hence, maximum queue rates limit the sustained data rate of a flow's service curve. In contrast, minimum queue rates enhance the service available to a flow by reducing the service

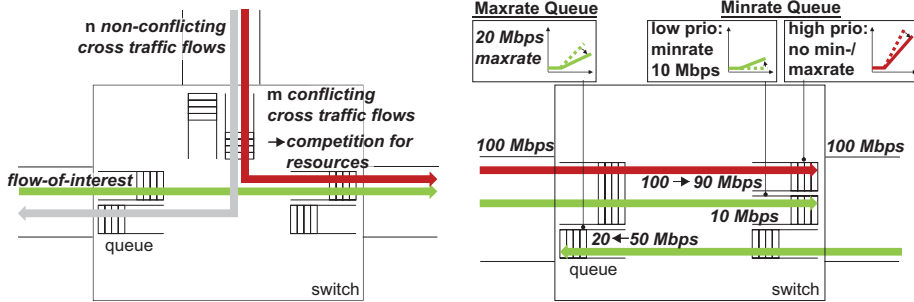


Figure 3: Extensions to Network Calculus: cross traffic handling and integration of queuing with minimum / maximum data rates

curves of higher priority flows, as shown by the right side of Figure 3. Equation 5 formalizes this concept for the service curve  $\beta_{k,foi}$  of a flow-of-interest  $foi$ ,

$$\beta_{k,foi}(t) = \beta_k(t) - \sum_{\forall q|p_q \geq p_{foi}} \left( \min \left( \sum_{\forall i \in q} \alpha_{k,i}(t), \alpha_{maxDR,q} \right) \right) - \sum_{\forall q|p_q < p_{foi} \cap \exists minDR_q} \left( \min \left( \sum_{\forall i \in q} \alpha_{k,i}(t), \alpha_{minDR,q} \right) \right), \quad (5)$$

with  $\beta_k$  being the basis service curve at node  $k$ . The service available to the flow-of-interest is reduced by the impact of traffic in queues  $q$  with same or higher priority ( $p_q \geq p_{foi}$ ), considering the sum of respective arrival curves  $\alpha_{k,i}$  of flows  $i$ . Yet, this influence may be limited by maximum queue rates  $\alpha_{maxDR,q}$ . Additionally, flows of lower priority ( $p_q < p_{foi}$ ) can curtail the service by up to the corresponding minimum queue rate  $\alpha_{minDR,q}$ .

To enable the analysis of non-feed forward networks, we enhance our modeling approach as illustrated by the left side of Figure 3. In classical NC such systems cannot be assessed as recursive calculation of cross traffic output curves may lead to deadlocks. This issue is avoided by considering only those cross traffic flows, which use the same output port as the flow-of-interest. We base this modification on the assumption that interference from other traffic flows at the switches' processing unit is negligibly small. This hypothesis is confirmed experimentally - for our testing environment - by the evaluations in Section 6.4.1. In

this way, analysis of cross-traffic in non-feed forward networks is converted back into a feed-forward problem. The associated definition of the left-over service curve  $\beta_{k,foi}$  for the flow-of-interest  $foi$  at node  $k$  is given by Equation 6,

$$\beta_{k,foi}(t) = \beta_k(t) - \sum_{\forall i | k_i+1=k_{foi}+1} \alpha_{k,i}(t), \quad (6)$$

where the node's basic service curve  $\beta_k$  is reduced by the arrival curves  $\alpha_{k,i}$  of cross traffic, which shares the same subsequent node  $k_i + 1$  as the flow-of-interest.

### 3.5.1. Network Calculus Application in the SDN Controller

Figure 4 gives an overview of the aims and different steps of these tasks. On the arrival of a new traffic flow, the SDN controller applies NC-based routing to select a delay-bound compliant path. We distinguish two different approaches for this task. Using the concept of full NC routing, the new flow's NC delay

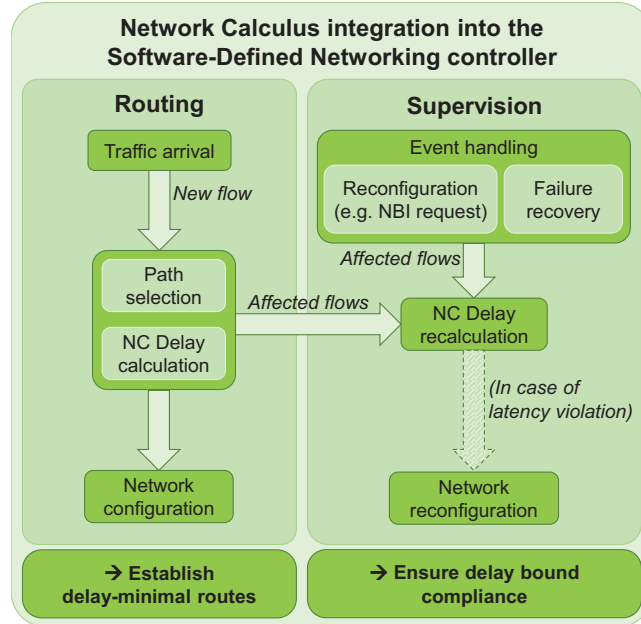


Figure 4: Concept for Network Calculus integration into the SDN controller

bounds are determined for every path provided by the DFS. Subsequently, the path with the lowest NC delay bound is chosen. In contrast, the hybrid NC routing approach couples standard service-aware routing and NC analysis. In this way, the delay-optimal path is selected by standard routing. Subsequently, the corresponding NC delay bound is calculated for this path only. If NC analysis does not indicate a potential violation of the given latency requirement, the selected route is configured in the network. Meanwhile, cross traffic, affected by the new flow, is handed over to NC delay supervision. In addition, delay supervision handles flows affected by network reconfiguration, for example in case of NBI-induced modified flow priorities or failure recovery. In all of the above cases NC delay bounds of affected traffic are recalculated and - if given latency requirements are exceeded - network reconfiguration is triggered. This involves measures such as rerouting and change of queues (priorities). For both, routing and delay supervision, performance can be enhanced by re-using previously calculated output bounds of respective cross traffic flows. Thus, calculations are sped up, whereas the recalculation of output bounds is not time critical and can be scheduled for subsequent execution. Detailed performance comparisons of the different routing approaches are provided in Section 6.4.3.

Algorithm 1 provides the main steps of our optimized NC delay analysis, which is applied for delay supervision and routing. The links of the intended path are iterated sequentially and checked for potential cross traffic (lines 2-3). To reduce computation times, previously computed output curves may be used for modeling cross traffic (lines 4-6). In case of non-optimized processing or if the curve has not been determined yet, recursive calculation of cross traffic output bounds - up to the point of interference - is required (lines 7-9). Next, cross traffic is classified with regard to its priority relative to the flow-of-interest and, if applicable, the service rate is bounded due to minimum/maximum queue rates (lines 10-20). Also, cross traffic flows are marked for output/delay bound recalculation as the flow-of-interest influences these flows vice versa (line 21). Afterwards, the base service curve for the flow-of-interest at the current node is retrieved (line 23). Cross traffic impact is determined according to Equation

---

**Algorithm 1:** Network Calculus Delay Supervision Algorithm

---

**Input:** Flow  $f$ , path  $p$   
**Result:** NC delay bound

```
1  $fPrio \leftarrow getPriority(f)$ 
2 for  $l$  in  $getLinksInPath(p)$  do
3   for  $cT$  in  $crossTraffic$  do
4     if  $outputCurves.contains(cT)$  then
5        $cToC \leftarrow getOutput(cT)$ 
6     end
7     else
8        $cToC \leftarrow computeOutputRecursive(cT)$ 
9     end
10    if  $getPrio(cT) > fPrio$  then
11       $cToC \leftarrow boundByMaxRate(cToC)$ 
12       $highLowPrio \leftarrow add(highLowPrio, cToC)$ 
13    end
14    else if  $getPrio(cT) < fPrio$  then
15       $cToC \leftarrow boundByMin(cToC)$ 
16       $highLowPrio \leftarrow add(highLowPrio, cToC)$ 
17    end
18    else
19       $samePrio \leftarrow add(samePrio, cToC)$ 
20    end
21     $markForRecalculation(cT, l)$ 
22  end
23   $sc \leftarrow getServiceCurve(f, l)$ 
24   $leftoverSC \leftarrow serviceCurve - highLowPrio$ 
25   $leftoverSC \leftarrow getFIFOService(sc, f, samePrio)$ 
26   $scETE \leftarrow convolve(scETE, leftoverSC)$ 
27 end
28  $ac \leftarrow getArrivalCurve(f)$ 
29  $delay \leftarrow getDelay(ac, scETE)$ 
30 for  $cT$  in  $markedDelayBounds$  do
31   if  $lastLatency(cT) + TH > maxLatency(cT)$  then
32      $recalculateDelay(cT)$ 
33   end
34 end
35  $scheduleRecalculation(markedOutBounds)$ 
```

---

6, using the corresponding output curves with respect to their relative priority  
500 (lines 24-25). By convolving individual service curves the end-to-end bound  
is calculated (Equation 3). The arrival curve, in conjunction with the end-  
to-end service curve, serves as input for deducing the flow-of-interest's upper  
delay bound (lines 28-29). Finally, delay bounds of critical flows, which are  
effected by the flow-of-interest, are recalculated (lines 30-34) and output bound  
505 recalculation is scheduled (line 35).

To sum up, priority-aware cross traffic modeling, refines delay computation  
and reflects actual network load. NC enables comprehensive evaluation of net-  
work performance, contrary to estimations gained from measurements.

#### 4. Smart Grid Reference Scenario and 510 Mapping on a Corresponding Communication Infrastructure

For evaluation we use the Nordic 32 test system [10], shown on the left side of  
Figure 5. The system, derived from actual Swedish and Nordic systems, is well-  
established for power grid analysis. It spans four voltage layers from 400 kV (red  
lines) to 15 kV (purple lines). On top of it, we map a corresponding wide-area  
515 communication network infrastructure, shown on the right side of Figure 5.  
Networking devices are placed at each substation and connected using fiber-  
optic cables, carried along the power lines. Thick, blue lines distinguish the  
area of the network, which is modeled in our empirical testbed setup, while the  
remaining part is emulated in Mininet [43] only. Figure 6 details our testbed  
520 implementation, while Section 5 provides hardware specifications.

To establish WAMPAC functionalities, we utilize IEC 61850 communica-  
tion services. In particular, SCADA-related control commands from the control  
center, situated at Substation 38, are sent to all substations using GOOSE mes-  
sages. SV serve for exchanging measurement data between neighboring substa-  
525 tions and with the control center. Starting from Subsection 6.2, MAS messaging  
is introduced for distributed power flow control within multiple clusters of sub-  
stations. Also, MMS transmissions are considered for configuration and software



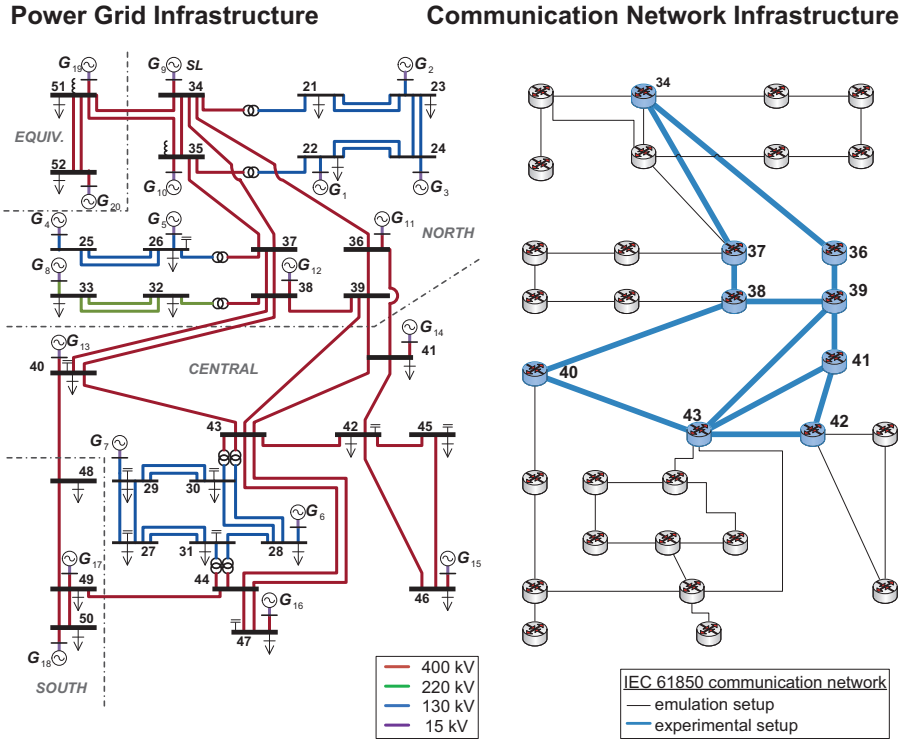


Figure 5: Mapping of the Nordic 32 Test System [10] for power grids to a corresponding IEC 61850-based ICT infrastructure

update purposes. Table 2 sums up used traffic patterns. In addition, Figure 6 visualizes the following sequence of use cases, considering GOOSE traffic from the control center (Substation 38) to Substation 41 as flow-of-interest for this analysis:

1. **Delay-aware routing** provides the primary path for this flow via Substations 38, 39, 41 (solid lines).
2. This path is interrupted by a **failure** between Substations 38 and 39, resulting in **recovery** to the fast (dashed lines) and the optimized failover path (dotted lines) (Section 6.1).
3. Evoked by the failure, combined with additional MAS and MMS traffic, the link between Substations 40 and 43 is **overloaded**. To maintain grid stability, dynamic **re-configuration** - triggered via the **NBI** - needs to

Table 2: Traffic patterns for Nordic 32 test system

Message Type	Source(s)	Destination(s)	Number of flows in experiment (simulation)	Scenarios (Sections)
GOOSE	38	all	8 (31)	1-4 (Sec. 6.1-6.4)
SV	all	38	8 (31)	1-4 (Sec. 6.1-6.4)
SV	all	neighbors	23 (85)	1-4 (Sec. 6.1-6.4)
MMS	38	34, 42	2 (8)	2 (Sec. 6.2)
MAS	38	41, 42, 43	3 (3)	2-4 (Sec. 6.2-6.4)
MAS	39	34, 36, 43	3 (3)	2-4 (Sec. 6.2-6.4)
MAS	(further MAS groups)		(17)	2-4 (Sec. 6.2-6.4)
<b>Total</b>			47 (178)	

540 be carried out (Section 6.2).

4. Finally, dash-dotted lines illustrate load optimization on basis of **multi-cast** transmission (Section 6.3).

## 5. Evaluation Environment for Empirical Performance Assessment

This section sums up the most important characteristics of our experimental environment as well as the used emulation software. Each experiment respec-  
 545 tively emulation is repeated 100 times with a duration of 60 s, typically resulting in up to 6 million data points per traffic flow.

### 5.1. Experimental Set-up

Our experimental environment, shown in Figure 6, consists of three inde-  
 550 pendent networks: data, control and management, created in hardware. The data network covers the data plane of the SDN architecture, representing the wide-area infrastructure for transmitting Smart Grid traffic. It includes four virtualized switches (vSwitch), running Open vSwitch (OVS) v2.5.2 under Ubuntu 16.04.2 LTS (v4.4.0-77-generic x86-64 Kernel) on standard server hardware (Intel Xeon D-1518 with one two port I210-LM and two four port I350 Intel 1GBase-T  
 555 Ethernet Network Interface Cards (NIC)). In addition, we deploy five 48 port

Pica8 3290 baremetal switches (bSwitch), which utilize OVS v2.3.0 under PicOS 2.6.32. The data network is completed by seven hosts, six of which are Intel Celeron J1900 with a two port I210-LM NIC. To achieve timing precision in the range of a few microseconds, while avoiding synchronization issues, the seventh host models Substations 38 and 41 simultaneously. Thus, corresponding measurements utilize a single clock.

The control network constitutes the SDN control plane, hosting the SUCCESS platform. Connection to the switches of the data plane is established using OpenFlow v1.3. Finally, the management network enables remote configuration, starting and stopping of measurement processes at all hosts. Hence, it is applied for facilitating the experiment and is not part of the evaluations itself. For both, the control and the management network, one Zyxel I2170-LM switch each provides Gigabit connectivity. Abstracting from real-world scenarios, copper instead of fiber-optic cables are employed.

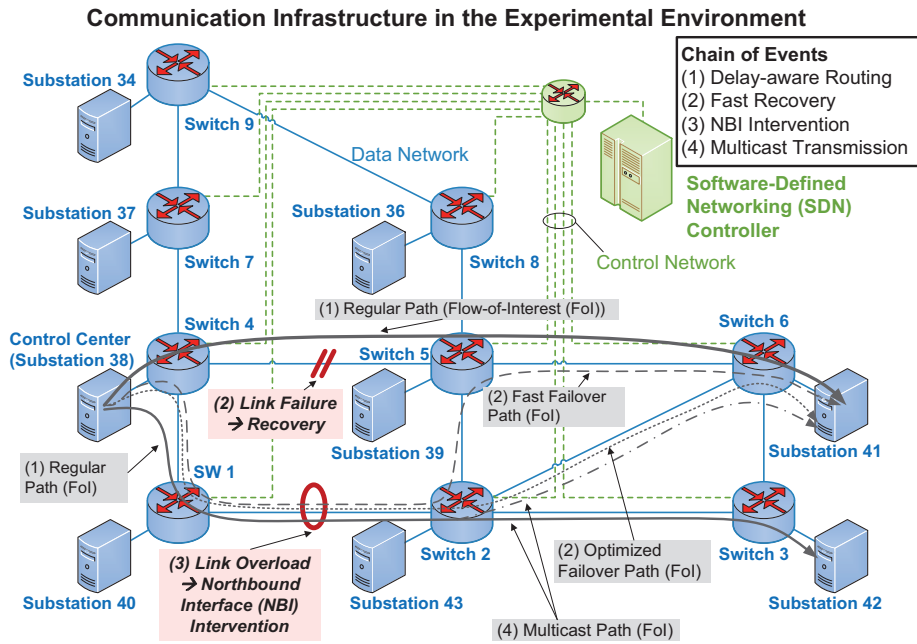


Figure 6: Experimental Realization of the Communication Network's Data and Control Plane Including Use Case Specific Paths of a Flow-of-Interest

## 5.2. Network Emulation

To validate the experimental results and scale evaluation to the full Nordic 32 test system, network emulations are carried out. Therefore, the software Mininet [43] is run on an Intel Xeon D-1518 under Ubuntu 16.04.2 LTS (v4.4.0-77-generic x86-64 Kernel). Mininet allows for the set-up of complex, realistic network configurations, applying the same controller framework as in the experiment. Configuration is performed using the Python programming language.

## 6. Evaluation of Approaches Proposed for Mission Critical Communications

Evaluation is split into four parts, each highlighting different hard service guarantee aspects, introduced in Section 3.

### 6.1. Comparison of Fast Failover Approaches

Within this subsection, we compare the failure detection and recovery mechanisms, described in Section 3, with regard to recovery delays, route optimality

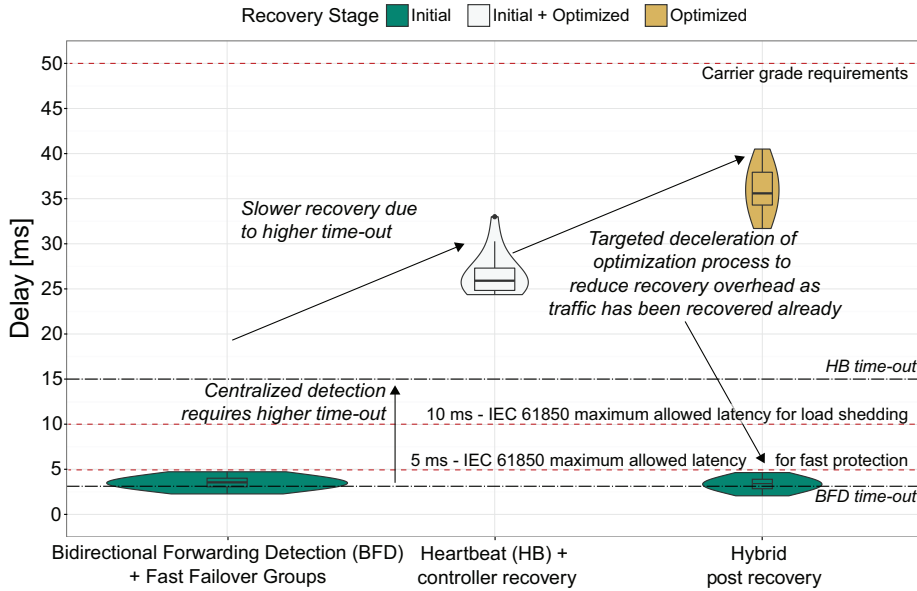


Figure 7: Comparison of Initial and Optimized Recovery Delay for Different Approaches using Software-Defined Networking

585 and induced link load on data and control network. Bidirectional Forwarding  
Detection (BFD) was configured with an Inter-Transmission Time (ITT) of 1 ms  
and a detection multiplier of 3, whereas the controller Heartbeat (HB) does not  
stabilize until an ITT of 3 ms, timing out after 15 ms. A link failure between  
Substations 38 and 39 is produced, interrupting the GOOSE traffic flow from  
590 the control center to Substation 41.

Figure 7 depicts the flow’s end-to-end recovery delays, measured at Substa-  
tion 41 in our testbed set-up (c.f. Figure 6). End-to-end recovery delay refers to  
the time difference between the last packet received before the failure and the  
first packet received after clearance. It can be seen that recovery delays depend  
595 significantly on the detection mechanism applied. Using BFD traffic is switched  
to an alternative path within 4.73 ms at maximum. In contrast, controller cen-  
tric failure detection and recovery requires up to 33 ms. Yet, this approach  
redirects the GOOSE traffic flow to an optimal path directly, whereas applying  
Fast Failover Groups (FFG) in combination with BFD necessitates subsequent  
600 optimization. This step may be triggered in response to the reception of regu-  
lar OFPortStatus messages, which is not until approximately 350 ms after the  
failure [9]. Integrating the advantages of both approaches, the hybrid approach  
uses BFD and FFG for immediate recovery, achieving the same latencies. In  
a second step Heartbeat (HB) messages are used to initialize controller-based  
605 post optimization with a mean delay of 35.94 ms, getting close to the recov-  
ery delay of the controller centric approach. To minimize network load of the  
hybrid approach, the HB interval for post optimization is increased to 10 ms.  
Post optimization delay can be reduced further, though at the cost of increased  
network load, due to more frequent HB messages.

610 Figure 8 illustrates the aspect of path optimality, considering the criteria  
minimum hop count (left side) and load balanced network links (right side).  
This study uses a Mininet emulation of the 75 link communication network of  
the full Nordic 32 system. The results of regular routing, before the failure,  
serve as benchmark for both cases (yellow boxes). The left-side of Figure 8  
615 visualizes the increase to a maximum hop count of eight due to FFG recovery.

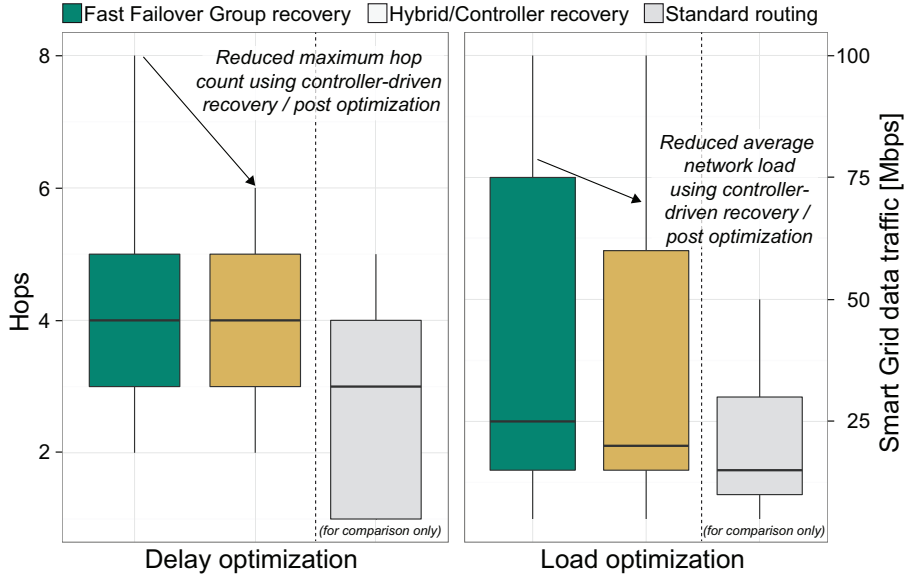


Figure 8: Hop counts and network load before/after failover using different Software-Defined Networking-enabled recovery methods

In comparison, the maximum hop count in case of controller recovery amounts to six only. According to the right-side of Figure 8, the median network load is reduced from 25 Mbps in case of FFG paths to 20 Mbps after controller recovery respectively post optimization.

620 Table 3 sums up the additional network load induced by the different failure detection mechanisms. These values are determined analytically for the monitoring of the whole 75 link Nordic 32 system. For this purpose, we assume a 10 Gbps control network, the ITT mentioned above and packet sizes

Table 3: Continuous additional load due to failure detection mechanisms on 75 data network links with 1 Gbps capacity each and 10 Gbps control network

Recovery Approach	Data Network	Control Network
BFD	0.019 %	0 %
Controller-Heartbeat	0.017 %	2.688 %
Hybrid	0.024 %	0.806 %
Hybrid optimized	0.019 %	0 %

according to [36]. While the controller HB achieves the lowest data network  
625 load of 0.017%, its frequent transmissions back to the SDN controller require  
2.688% of the control network capacity, which is the highest demand among  
all approaches. In comparison, even the hybrid approach, which comprises less  
frequent HB messages, incurs a control network load of just 0.806%. However,  
a slight increase in data network load to 0.024% has to be noted. Finally, the  
630 data network load of BFD is in between the other two approaches, whereas the  
control network is only stressed in case of failure. Further optimization of the  
hybrid mechanism, may reduce its associated network loads to the same levels  
as those of BFD. While the load on the monitored link is comparatively low  
in all cases ( $< 0.1\%$ ), the control network could experience considerable stress,  
635 depending on control network topology and number of monitored links. In ad-  
dition, assuming adequate processing resources being available to the controller,  
it needs to be highlighted that scalability of the recovery approaches boils down  
to the issue of control network utilization.

All in all, the hybrid recovery concept can be considered a reasonable com-  
640 promise between low recovery delays, path optimality and consumed network  
capacity, the latter one even being improved by an optimized version of the  
approach.

### 6.2. Smart Grid Service-Driven Dynamic Priority Adaption

Using the example of varying service requirements for Multi-Agent Sys-  
645 tem (MAS)-based distributed power grid control, dynamic adaption of network  
configurations is shown. This involves prioritization, queuing and Northbound  
Interface (NBI) requests. A five step sequence of dynamic prioritization tasks is  
executed, as shown in Figure 9. The sequence involves two of the NBI requests,  
introduced in Section 3.

650 In step 1, MAS traffic is transmitted on an empty link between Switches 40  
and 43. In total, these MAS messages have a capacity demand of approximately  
5 Mbps, illustrated by bar plots in the upper part of Figure 9. This results in  
mean latencies of  $351 \mu\text{s}$ , depicted by the violin plots in the graph's lower part.

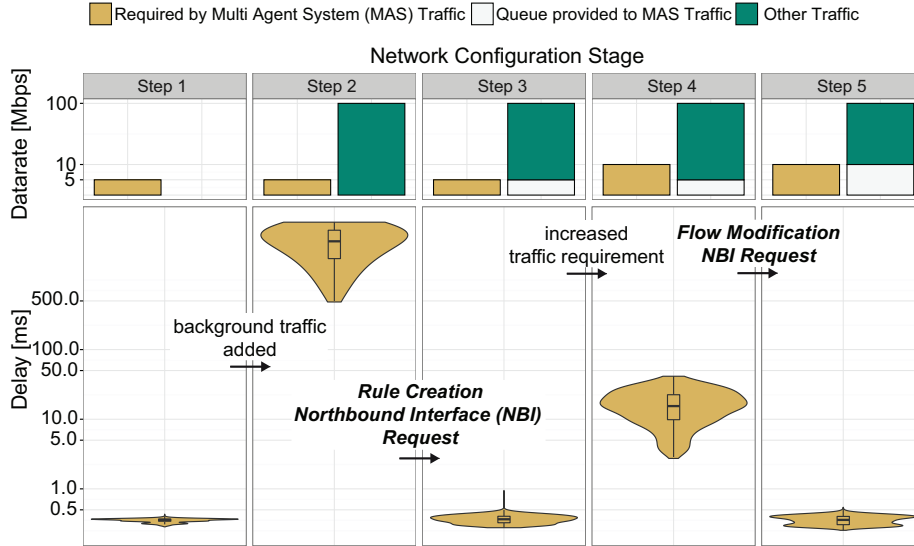


Figure 9: Successive steps of handling Multi-Agent System (MAS) traffic in response to changing network conditions, Northbound Interface (NBI) requests and subsequent priority/queue assignment

Next, normal traffic conditions, as described in Section 4, are restored with  
655 GOOSE and SV traffic being present on the network as well. Further, additional MMS traffic for the purpose of updating devices is injected into the ICT infrastructure. In conjunction with the link failure, discussed in the previous subsection, this leads to an overload of the link between Substations 40 and 43 as shown in step 2 of Figure 9. Since MAS traffic is not recognized by the  
660 controller yet, it is handled as best effort, causing a drastic increase of the delay of up to 6.76 s. To resolve this issue, a Rule Creation request is sent. Thus, the MAS priority is raised to 30, which is well above the priority of MMS (priority level 20). Adequate queues with 5 Mbps minimum data rate are arranged for. Hence, delays are reduced back to below 1 ms, as shown in step 3. Next,  
665 due to the power system being highly loaded and not in (N-1) secure state, an outage occurs, disconnecting the transmission line between Substations 38 and 39. Subsequently, parallel transmission lines between Substation 40 and 43 become overloaded. This emergency situation is identified by the agents of the distributed control system. To prevent cascading outages, the MAS aims at



670 estimating the grid state on basis of refined measurement data. Accordingly, its  
monitoring precision has to be improved. Building on the detailed view of the  
power system, adequate counter-measures can be determined, which - in this  
case - involves triggering a Power Flow Controller (PFC). These developments  
lead to more frequent transmissions of critical MAS messages, thus increasing  
675 the traffic load, as shown in step 4 of Figure 9. However, the queue assigned to  
MAS messaging is not sufficient for these altered data rate requirements, caus-  
ing a rise in delay up to 41.43 ms. Subsequently, a Flow Modification request  
is issued to obtain a temporary raise of priority. Thus, MAS traffic is switched  
to a higher priority queue, providing up to 10 Mbps minimum data rate and  
680 restoring the initial delay level (step 5 of Figure 9). In this way, despite of the  
heavily loaded communication network, timely transmission of critical control  
messages can be ensured. In turn, power system stability can be maintained,  
preventing cascading outages.

### 6.3. Validation of Multicast Load Reduction

685 This subsection targets load reduction with the means of multicast transmis-  
sion. Therefore, transfer of measurement values or statuses from one to multiple  
other substations is bundled in multicast transmissions, wherever possible. In  
addition, if identical commands are sent by the control center to several sub-  
stations, these GOOSE messages are transferred as multicast. On shared paths  
690 between different agents of the distributed control system joint transmission is  
employed as well. The resulting optimization of bandwidth consumption in the  
network is studied using both emulation and experiment.

Figure 10 (left side) contrasts network utilization for unicast and multicast  
transmission, measured in our testbed. Compared to unicast, the mean link  
695 load is reduced from 7.50 Mbps to 6.63 Mbps. In addition, applying multicast  
diminishes the maximum load by 11.1 % to 15.47 Mbps, shown by marker (a).

Scaling up, emulation is used to study the impact of deploying multicast on  
the whole 75 link Nordic 32 system. While in the experiment we focus on delay  
optimal routing, the simulation includes load optimal routing as well. Figure 10

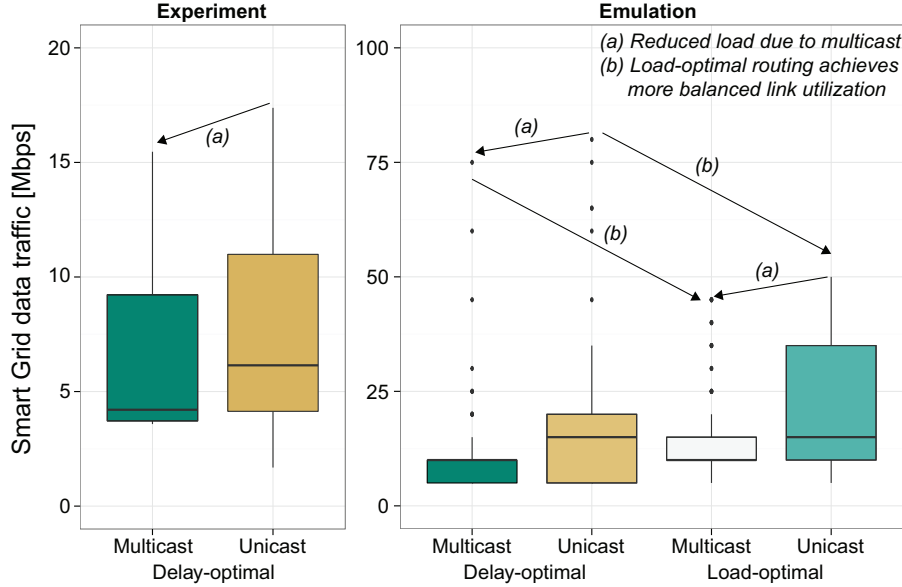


Figure 10: Comparison of network load using uni- and multicast flows in simulation (left) and experiment (right)

700 (right side) shows link loads for the four different combinations of uni-/multicast transmission and delay/load optimal routing. Similar to the experiment a reduction of mean and maximum load is observed, when exchanging unicast for multicast transfers, highlighted by marker (a). This holds true for both routing disciplines. Comparing the different routing schemes - among each pair of  
 705 unicast respectively multicast transmissions - shows an increase of mean link utilization for load optimal routing, whereas the maximum load is delimited to a lower level as can be seen from marker (b). This behavior matches perfectly the concept of balancing network utilization.

#### 6.4. Evaluation of In-Controller Network Calculus Supervision and Routing

710 As described in Section 3.4, we apply Network Calculus (NC) for delay-aware routing of traffic flows and online supervision of latency requirement compliance. In the following, prerequisite evaluations are performed for assuring the assumptions of modified cross traffic handling. Next, calculated NC delay bounds

are cross-validated against the results of empirical measurements. This section concludes with evaluations on the applicability of NC-based routing and delay supervision.

#### 6.4.1. Prerequisite Assessment of Cross Traffic Handling

Preliminary studies for NC application include the analysis of switching delays of a virtualized switch for different ITT and traffic conditions, as illustrated in Figure 11. It needs to be stressed that these evaluations only serve for conforming the assumptions on cross traffic behavior described in Section 3.4 and do not reflect actual traffic configurations considered in the remainder of this article.

To deduce latencies, traffic captures of one specific flow-of-interest at the ingress and egress port of the switch are considered. The single traffic flow case constitutes a scenario, in which only the flow-of-interest is present, whereas in the cross traffic case a second flow uses the same egress port. In contrast, in the full traffic load scenario additional communication streams reach the

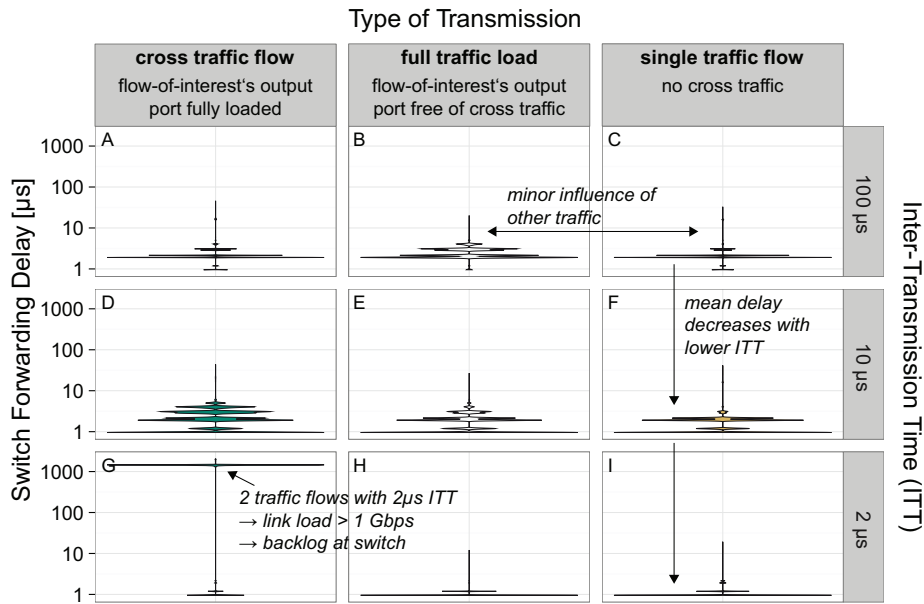


Figure 11: Traffic flow switching delay for different Inter-Transmission Time (ITT) and (cross) traffic conditions on a 1 Gbps network

switch, however obviating the egress port used by the flow-of-interest. It can be  
730 observed that delay decreases with reduced ITT (for a more detailed analysis  
of this phenomenon c.f. [44]). Meanwhile, additional traffic at the switch shows  
minimal influence on the switching performance, if different egress ports are  
used. In comparison, cross traffic being present on the same egress port, evokes  
rising delays of the flow-of-interest. If the competing traffic flows exceed the  
735 maximum capacity of the connected egress link, which is true for an ITT of  
 $2\ \mu\text{s}$ , delay even increases by three orders of magnitude (c.f. Figure 11.G).  
Accordingly, traffic using the same output port as the flow-of-interest needs  
to be considered for delay analysis due to its significant impact, whereas the  
influence of traffic flows on other output ports has been shown to be negligible.  
740 Hence, NC can be simplified in this regard, as described in Section 3.4. This  
obviates the issue of looped flow dependencies, which otherwise might cause  
deadlocks in computation. On the other hand, measurements reveal the need  
for considering the impact of varying ITT on switching latencies. Subsequently,  
these findings are integrated into NC.

#### 745 6.4.2. Validation of Network Calculus Delay Bounds

In the next step, we aim at comparing measured network delays to the results  
of NC-based flow analysis in order to prove its applicability for network state  
monitoring and delay supervision. Figure 12 comprises measured delays in terms  
of violin and box plots for GOOSE and MAS transmissions between the control  
750 center (Substation 8) and Substation 41, considering three different scenarios.  
Above the violins dotted lines indicate the maximum measured delay, whereas  
solid lines represent the corresponding results of in-controller NC analysis. In  
comparison to the previous evaluation, the traffic loads listed in Table 2 are  
restored. Hence, the two flows-of-interest are interfered by multiple cross traffic  
755 flows.

The scenarios considered map to the use cases presented in the course of  
this paper: *before failure* of the communication link between Substations 38  
and 39, *after failure recovery* to alternative paths and *after applying multicast*

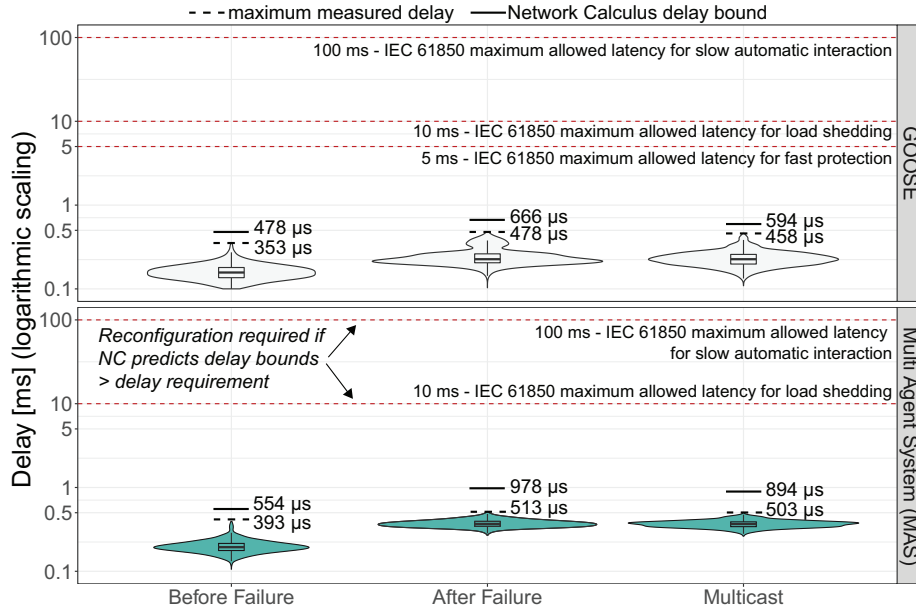


Figure 12: Measured delays (violin and box plots) and Network Calculus (NC) bounds (solid lines) of GOOSE and Multi-Agent System (MAS) traffic from Host 4 to 6 for different scenarios

transmission mode. Dynamic prioritization is excluded here, since it would  
 760 involve overloading communication links, resulting in infinite delay bounds in  
 NC.

In all three scenarios, NC bounds are not exceeded, being 120 to  $450\ \mu\text{s}$   
 above the maximum values, measured in the testbed. Deviation between NC  
 bounds and maximum measured values increases for the case of MAS traffic  
 765 after occurrence of the ICT failure. This effect can be attributed to NC's sensitivity  
 to prioritization. In this case, the behavior is sparked by relatively low priority  
 of the MAS service in combination with numerous - higher priority - cross traffic  
 flows, being present on the back-up route. Nevertheless, evaluation highlights  
 that NC provides valid means of network latency estimation within SUCCESS.  
 770 Delay bounds are found to be well-above maximum measurement results, while  
 not being overly loose.

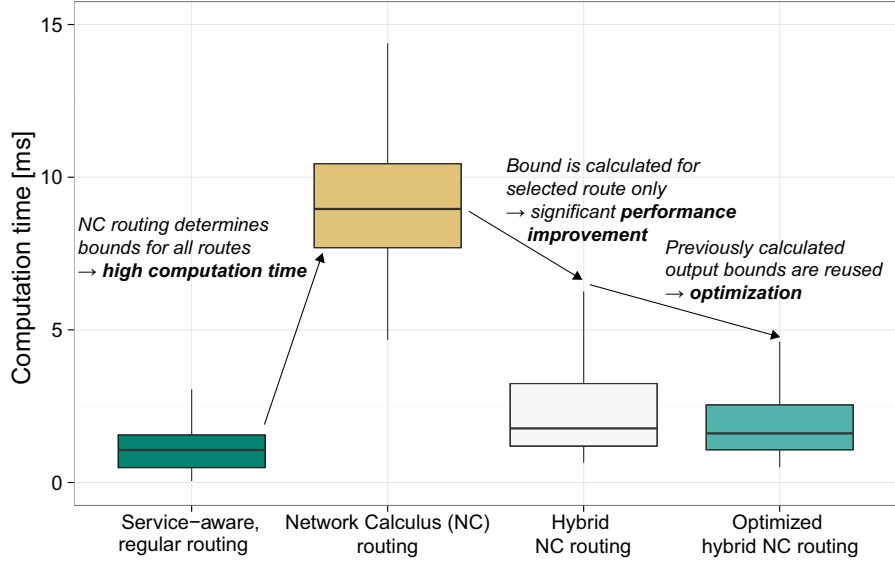


Figure 13: Comparison of computation times for regular, Network Calculus-based and hybrid routing approaches, used in our Software-Defined Networking (SDN) Controller

#### 6.4.3. Evaluation of Network Calculus-based Routing

Figure 13 compares the performance of NC based routing with the computation times of regular, service-aware routing approach. While the regular routing  
 775 completes within less than 3 ms at maximum, full NC-based routing incurs mean delays of 14.44 ms. Computation speed of this NC routing approach is determined by the fact, that delay bounds are derived for all feasible routes within the full Nordic 32 communication network. Further performance improvements may be achieved by parallelizing these calculations. In contrast, the hybrid  
 780 NC routing concept builds on the idea of coupling service-aware routing and NC analysis. Therefore, an optimal route is determined using regular routing, for which delay bound compliance is checked with the help of NC. Hence, performance is improved to mean computation times of 2.66 ms. To further optimize computation times of NC routing, we re-use previously calculated output  
 785 bounds during delay bound calculation for the new flow-of-interest as described in Algorithm 1. This obviates efforts of recursively determining output bounds on-the-fly. Subsequently, the mean calculation period is decreased to 2.17 ms in

case of optimized hybrid NC routing, however at the cost of reduced precision of the delay bound.

790 6.4.4. Optimization of Network Calculus Computation Times

The following evaluation focuses on the optimization of NC computation times for the application within the SDN controller. A detailed performance comparison of the standard algorithm and the optimized approach is provided in Figure 14, displaying empirical measurements of execution durations. The standard algorithm was utilized for NC and hybrid NC routing, whereas the enhanced version has been employed for optimized hybrid NC routing as well as NC delay supervision. Following the standard approach, output bounds of all cross traffic flows are computed on-the-fly during delay calculation of the flow-of-interest (first column), leading to maximum computation times of 76 ms. 795 Afterwards, the delay of all previously installed traffic flows is recalculated, considering the impact of the new flow (second column). This step may take up to approximately 1 s. Initial delay analysis of the flow-of-interest can be sped up by making use of previously calculated output bounds. Thus, calculation times

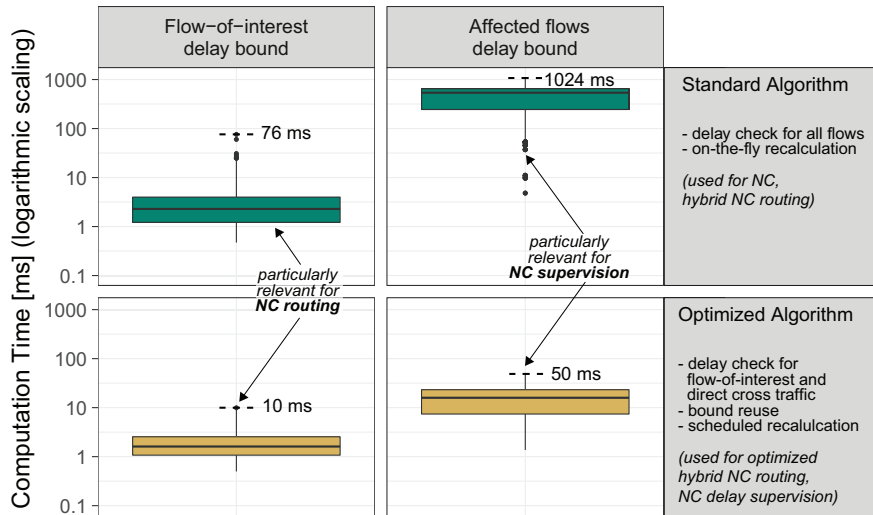


Figure 14: Comparison of computations times for different calculation objects and algorithms with relevant parameters for NC routing respectively NC delay supervision being highlighted

can be reduced to maxima of 10 ms for the flow-of-interest and 50 ms for affected  
805 cross traffic flows. The latter provides a worst-case estimation as delay bounds  
for all cross traffic flows are recomputed. In real-world scenarios it would be  
sufficient to recalculate the delay bounds of those flows close to their respec-  
tive latency requirements. Yet, due to the concept of reusing existing output  
bounds, it becomes necessary to perform a third calculation step, recalculating  
810 the output bounds. Nevertheless, this final step does not need to be executed  
immediately, but may be scheduled.

#### 6.4.5. Assessment of Delay Supervision for Dynamic Reconfiguration

Finally, the application of NC delay supervision in the context of dynamic  
network reconfiguration is evaluated. As shown in Figure 4, reconfiguration may  
815 be caused by the insertion of new traffic flows, as direct and indirect result of  
NBI requests or evoked by failure recovery. In this context, Figure 15 comprises  
measurement results for the delay of network reconfiguration in terms of a violin  
and overlaid box plot. The median reconfiguration time amounts to 3.37 ms,  
whereas at maximum delays of 6.12 ms are reached. These reconfiguration times  
820 are taken into account for subsequent analysis. Table 4 focuses on the case  
of NBI request-induced network reconfiguration, comparing delay impacts of

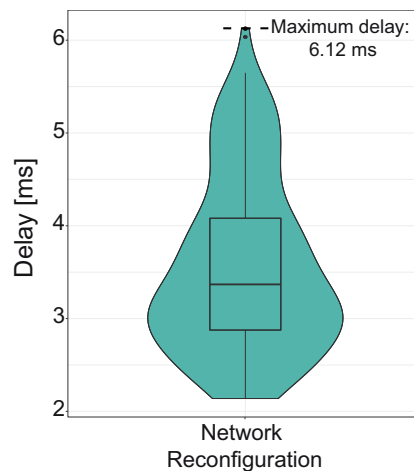


Figure 15: Measured network reconfiguration delays



Table 4: Delay impact of computation times derived from the results presented in Figures 14 and 15

Options	Chain of Events	Delay impact [ms]			
		Request. flow		Affected flows	
		Mean	Max	Mean	Max
Post-reconfiguration check	1. Request	4	6	-	-
	2. Reconfiguration of requesting flow	4	6	-	-
	3. NC recalculation	-	-	17	49
	4. Reconfiguration of affected flows	-	-	4	6
	<b>Total</b>	8	<b>12</b>	14	<b>56</b>
→ in the worst case, <b>affected flows</b> impacted considerably					
Pre-reconfiguration check	1. Request	4	6	-	-
	2. NC recalculation	17	49	-	-
	3. Reconfiguration of affected flows	4	6	-	-
	4. Reconfiguration of requesting flow	4	6	-	-
	<b>Total</b>	22	<b>68</b>	0	<b>0</b>
→ in the worst case <b>requesting flow</b> impacted considerably					
→ applicable for <b>Smart Grid</b> services with <b>latency requirements</b> $\geq 100$ ms					

different implementation options. These alternatives deviate with regard to the order, in which processes are executed. In case of *post-reconfiguration check* the network configuration (queue rate, priority) is altered immediately, resulting in maximum adjustment latencies of about 12 ms for the requesting flow. Only afterwards, NC is employed to recalculate the delay bounds of affected flows and check for potential violations of given latency requirements. If so, subsequent reconfiguration of the affected traffic flows has to be performed. Accumulating NC computation and corresponding reconfiguration times, a worst case delay of 56 ms is constituted. In contrast, using the *pre-reconfiguration check* other flows are not influenced by the NBI request as potential effects on their delay bounds is assessed beforehand. However, in this way the reconfiguration of the requesting

flow is delayed by up to 68 ms. Hence, both approaches exhibit advantages and disadvantages, either for the requesting flow or for affected transmissions. Taking into account Smart Grid latency requirements defined in Table 1, the following conclusions can be drawn:

- Combining NC delay supervision with dynamic network reconfiguration allows for flexibly reallocating resources for Smart Grid traffic flows with latency requirements  $\geq 100$  ms as delay compliance is ensured at all times.
- In contrast, extremely time critical services with latency requirements  $< 10$  ms may not be subjected to reconfiguration at any time.
- Vice versa, minimum and maximum queue concepts have to be employed for assuring dedicated resources for these services. Respective configurations must not be altered during failover or reconfiguration.
- Further optimization of algorithms and hardware set-up may enable extending dynamic, NC monitored network reconfiguration to Smart Grid services with latency requirements of 10-100 ms.

## 7. Conclusion and Future Work

To cope with the complex challenges of mission critical communications in cyber-physical systems, we proposed the use of Software-Defined Networking (SDN) on basis of our Software-Defined Universal Controller for Communications in Essential Systems (SUCCESS) framework. In this article we focused on the case of emerging Smart Grid infrastructures, evaluating the suitability of our approach with the help of experiments and emulations. Therefore we modeled an ICT infrastructure on top of the well-established Nordic 32 test system and derived specific scenarios for each aspect of hard service guarantees. Reliability of communication networks was studied with regard to handling critical link failures. Applying a hybrid concept, combining distributed and centralized failure detection and recovery, maximum delays of 5 ms are achieved, while

860 maintaining optimal paths almost continuously. Dynamic adaptation of priorities (queues) is utilized for minimizing communication delays of a Multi-Agent System (MAS) even in the presence of high traffic load. Alternating requirements are conveyed via the controller's Northbound Interface (NBI), relying on the REST API. In addition, the NBI is used for creating multicast groups, as  
865 commonly used in IEC 61850 communications, significantly reducing average and maximum link load. Finally, the analytical modeling approach of Network Calculus (NC) was integrated into SUCCESS and tailored to the specifics of min/max rate queuing as implemented at the switches within our testing environment. Hence, real-time capability of critical communications can be monitored online on basis of hard worst case delay bounds. In case of violations,  
870 remedial actions, such as fast re-routing or dynamic priority adaptation, are applied. In contrast to measurement-based latency supervision, NC integration enables a comprehensive view on delays, their triggers and even predictions of future endangerments. Yet, we also indicated limits of NC-monitored dynamic  
875 network reconfiguration as - for numerous traffic flows - computation times may jeopardize latency requirements of extremely time critical Smart Grid protection functions ( $<10$  ms). Further, NC was utilized for improved, delay-bounded routing.

Further enhancing our reliability concept, subsequent work will deal with  
880 fast failure recovery for multicast traffic flows. Moreover, we aim at establishing communication between distributed, inter-connected controllers in order to achieve a) controller resilience and b) improve the scalability. Additionally, assignment of transmission capacities in wireless networks can be added to the controller's capabilities.

## 885 **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  
890 and Research (BMBF).

## References

- [1] X. Fang, S. Misra, G. Xue, D. Yang, Smart Grid - The New and Improved Power Grid: A Survey, *IEEE Communications Surveys and Tutorials* 14 (4) (2012) 944–980. doi:10.1109/SURV.2011.101911.00087.
- 895 [2] Y. Yan, Y. Qian, H. Sharif, D. Tipper, A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges, *IEEE Communications Surveys and Tutorials* 15 (1) (2013) 5–20. doi:10.1109/SURV.2012.021312.00034.
- [3] N. S. Nafi, K. Ahmed, M. A. Gregory, M. Datta, A survey of smart grid architectures, applications, benefits and standardization, *Journal of Network and*  
900 *Computer Applications* 76 (2016) 23 – 36. doi:10.1016/j.jnca.2016.10.003.
- [4] International Electrotechnical Commission TC57, IEC 61850: Communication Networks and Systems for Power Utility Automation.
- [5] M. Albano, L. L. Ferreira, L. M. Pinho, Convergence of Smart Grid ICT Architectures for the Last Mile, *IEEE Transactions on Industrial Informatics* 11 (1)  
905 (2015) 187–197. doi:10.1109/TII.2014.2379436.
- [6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: Enabling Innovation in Campus Networks, *SIGCOMM Computer Communication Review* 38 (2) (2008) 69–74. doi:  
910 10.1145/1355734.1355746.
- [7] A. Sydney, J. Nutaro, C. Scoglio, D. Gruenbacher, N. Schulz, Simulative Comparison of Multiprotocol Label Switching and OpenFlow Network Technologies for Transmission Operations, *Transactions on Smart Grids* 4 (2) (2013) 763–770. doi:10.1109/TSG.2012.2227516.
- 915 [8] J. Le Boudec, P. Thiran, *Network Calculus - A Theory of Deterministic Queuing Systems for the Internet*, Springer, 2004. doi:10.1007/3-540-45318-0.

- [9] N. Dorsch, F. Kurtz, H. Georg, C. Hägerling, C. Wietfeld, Software-Defined Networking for Smart Grid Communications: Applications, Challenges and Advantages, in: IEEE International Conference on Smart Grid Communications, 2014, pp. 422–427. doi:10.1109/SmartGridComm.2014.7007683.
- [10] CIGRE Task Force 38.02.08, Long Term Dynamics Part II - A practical Assessment of Simulation Tools (1995).
- [11] M. Kuzlu, M. Pipattanasomporn, S. Rahman, Communication network requirements for major smart grid applications in HAN, NAN and WAN, Computer Networks 67 (2014) 74 – 88. doi:https://doi.org/10.1016/j.comnet.2014.03.029.
- [12] S. C. Müller, U. Häger, C. Rehtanz, A Multi-Agent System for Adaptive Power Flow Control in Electrical Transmission Systems, IEEE Transactions on Industrial Informatics 10 (4) (2014) 2290–2299. doi:10.1109/TII.2014.2315499.
- [13] N. Dorsch, F. Kurtz, S. Dalhues, L. Robitzky, U. Häger, C. Wietfeld, Intertwined: Software-Defined Communication Networks for Multi-Agent System-based Smart Grid Control, in: 2016 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2016, pp. 254–259. doi:10.1109/SmartGridComm.2016.7778770.
- [14] Open Networking Foundation, OpenFlow Switch Specification Version 1.3.0 (2012).  
URL <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf>
- [15] H. Farhady, H. Lee, A. Nakao, Software-Defined Networking: A survey, Computer Networks 81 (2015) 79 – 95. doi:https://doi.org/10.1016/j.comnet.2015.02.014.
- [16] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-Defined Networking: A Comprehensive Survey, Proceedings of the IEEE 103 (1) (2015) 14–76. doi:10.1109/JPROC.2014.2371999.
- [17] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, A. Vahdat, B4: Ex-

- perience with a Globally-deployed Software Defined WAN, SIGCOMM Comput. Commun. Rev. 43 (4) (2013) 3–14. doi:10.1145/2534169.2486019.
- [18] A. Rostami, P. Ohlen, K. Wang, Z. Ghebretensae, B. Skubic, M. Santos, A. Vidal, 950  
Orchestration of RAN and Transport Networks for 5G: An SDN Approach, IEEE Communications Magazine 55 (4) (2017) 64–70. doi:10.1109/MCOM.2017.1600119.
- [19] R. L. Cruz, A calculus for network delay. I. Network elements in isolation, IEEE Transactions on Information Theory 37 (1) (1991) 114–131. doi:10.1109/18. 955  
61109.
- [20] J. W. Guck, M. Reisslein, W. Kellerer, Function Split Between Delay-Constrained Routing and Resource Allocation for Centrally Managed QoS in Industrial Networks, IEEE Transactions on Industrial Informatics 12 (6) (2016) 2050–2061. doi:10.1109/TII.2016.2592481.
- [21] Q. Duan, Network-as-a-Service in Software-Defined Networks for end-to-end QoS 960  
provisioning, in: 2014 23rd Wireless and Optical Communication Conference (WOCC), 2014, pp. 1–5. doi:10.1109/WOCC.2014.6839919.
- [22] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, N. Venkatasubramanian, A Software Defined Networking architecture for the Internet-of-Things, in: 2014 965  
IEEE Network Operations and Management Symposium (NOMS), 2014, pp. 1–9. doi:10.1109/NOMS.2014.6838365.
- [23] S. Azodolmolky, R. Nejabati, M. Pazouki, P. Wieder, R. Yahyapour, D. Simeonidou, An analytical model for software defined networking: A network calculus-based approach, in: 2013 IEEE Global Comm. Conf. (GLOBECOM), 2013, pp. 970  
1397–1402. doi:10.1109/GLocom.2013.6831269.
- [24] J. Huang, L. Xu, Q. Duan, C. cong Xing, J. Luo, S. Yu, Modeling and performance analysis for multimedia data flows scheduling in software defined networks, Journal of Network and Computer Applications 83 (2017) 89 – 100. doi:10.1016/j.jnca.2017.01.032.
- [25] A. K. Koohanestani, A. G. Osgouei, H. Saidi, A. Fanian, An analytical model for 975  
delay bound of OpenFlow based SDN using network calculus, Journal of Network

and Computer Applications 96 (2017) 31 – 38. doi:10.1016/j.jnca.2017.08.002.

- 980 [26] N. Dorsch, H. Georg, C. Wietfeld, Analysing the Real-Time-Capability of Wide Area Communication in Smart Grids, in: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2014, pp. 682–687. doi:10.1109/INFCOMW.2014.6849313.
- 985 [27] E. Molina, E. Jacob, Software-defined networking in cyber-physical systems: A survey, Computers & Electrical Engineering 66 (2018) 407 – 419. doi:10.1016/j.compeleceng.2017.05.013.
- [28] A. Cahn, J. Hoyos, M. Hulse, E. Keller, Software-Defined Energy Communication Networks: From Substation Automation to Future Smart Grids, in: IEEE International Conference on Smart Grid Communications, 2013, pp. 558–563. doi:10.1109/SmartGridComm.2013.6688017.
- 990 [29] E. Molina, E. Jacob, J. Matias, N. Moreira, A. Astarloa, Using Software Defined Networking to manage and control IEC 61850-based systems, Computers and Electrical Engineering 43 (2015) 142 – 154. doi:10.1016/j.compeleceng.2014.10.016.
- 995 [30] E. G. d. Silva, A. S. d. Silva, J. A. Wickboldt, P. Smith, L. Z. Granville, A. Schaeffer-Filho, A One-Class NIDS for SDN-Based SCADA Systems, in: 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Vol. 1, 2016, pp. 303–312. doi:10.1109/COMPSAC.2016.32.
- 1000 [31] S. Wang, X. Huang, Aggregation points planning for software-defined network based smart grid communications, in: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1–9. doi:10.1109/INFOCOM.2016.7524564.
- 1005 [32] L. Ren, Y. Qin, Y. Li, P. Zhang, B. Wang, P. B. Luh, S. Han, T. Orekan, T. Gong, Enabling resilient distributed power sharing in networked microgrids through software defined networking, Applied Energy 210 (2018) 1251 – 1265. doi:10.1016/j.apenergy.2017.06.006.

- [33] A. Sydney, D. S. Ochs, C. Scoglio, D. Gruenbacher, R. Miller, Using GENI for experimental evaluation of Software Defined Networking in smart grids, *Computer Networks* 63 (2014) 5 – 16, Special Issue on Future Internet Testbeds - Part II. doi:10.1016/j.bjp.2013.12.021.
- 1010 [34] A. Aydeger, K. Akkaya, M. H. Cintuglu, A. S. Uluagac, O. Mohammed, Software defined networking for resilient communications in Smart Grid active distribution networks, in: 2016 IEEE International Conference on Communications (ICC), 2016, pp. 1–6. doi:10.1109/ICC.2016.7511049.
- 1015 [35] T. Pfeiffenberger, J. L. Du, P. B. Arruda, A. Anzaloni, Reliable and flexible communications for power systems: Fault-tolerant multicast with SDN/OpenFlow, in: International Conference on New Technologies, Mobility and Security, 2015, pp. 1–6. doi:10.1109/NTMS.2015.7266517.
- [36] N. Dorsch, F. Kurtz, F. Girke, C. Wietfeld, Enhanced Fast Failover for Software-Defined Smart Grid Communication Networks, in: IEEE Global Comm. Conf. (GLOBECOM), 2016, pp. 1–6. doi:10.1109/GLOCOM.2016.7841813.
- 1020 [37] D. Ivancic, N. Hadjina, D. Basch, Analysis of precision of the HTB packet scheduler, in: 18th International Conference on Applied Electromagnetics and Communications, 2005, pp. 1–4. doi:10.1109/ICECOM.2005.204958.
- [38] International Electrotechnical Commission (IEC) TC57, IEC 61850 Communication networks and systems for power utility automation Part 90-5: Use of IEC 61850 to transmit synchrophasor information according to IEEE C37.118.
- 1025 [39] D. Katz, D. Ward, Bidirectional Forwarding Detection (BFD) (RFC 5880), Tech. rep., Internet Engineering Task Force (IETF) (June 2010). doi:10.17487/RFC5880.
- 1030 [40] Open vSwitch Version 2.5.2/2.3.0, Accessed: 30 January 2018 (2015). URL <http://openvswitch.org/>
- [41] M. Tacca, K. Wu, A. Fumagalli, J. p. Vasseur, Local Detection and Recovery from Multi-Failure Patterns in MPLS-TE Networks, in: IEEE International Conference on Communications, Vol. 2, 2006, pp. 658–663. doi:10.1109/ICC.2006.254782.



- 1035 [42] M. Parsa, Q. Zhu, J. J. Garcia-Luna-Aceves, An iterative algorithm for delay-constrained minimum-cost multicasting, *IEEE/ACM Transactions on Networking* 6 (4) (1998) 461–474. doi:10.1109/90.720901.
- [43] Mininet. An Instant Virtual Network on your Laptop (2014).  
URL <http://mininet.org>
- 1040 [44] F. Kurtz, N. Dorsch, C. Wietfeld, Empirical Comparison of Virtualised and Bare-Metal Switching for SDN-based 5G Communication in Critical Infrastructures, in: *IEEE Conference on Network Softwarization*, 2016, pp. 453–458. doi:10.1109/NETSOFT.2016.7502484.