Accepted Manuscript

Software defined network management for dynamic smart GRID traffic

Mita Cokic, Ivan Seskar

PII: S0167-739X(18)31106-3
DOI: https://doi.org/10.1016/j.future.2019.02.022
Reference: FUTURE 4775

To appear in: *Future Generation Computer Systems*

Received date : 10 May 2018
Revised date : 4 November 2018
Accepted date : 13 February 2019

Please cite this article as: M. Cokic and I. Seskar, Software defined network management for dynamic smart GRID traffic, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2019.02.022

Software Defined Network Management for Dynamic Smart Grid Traffic

Mita Cokic[a,*], Ivan Seskar[b]

[a] *Faculty of Technical Sciences, Unversity of Novi Sad, Trg Dositeja Obradovica 6, Novi Sad 21000, Serbia*
[b] *WINLAB, Rutgers University, 671 US1 South, North Brunswick, NJ 08902, USA*

Abstract

One of the more challenging issues in Smart Grid (SG) communications is in handling the ever-increasing number of new SG applications that are being provisioned by the utility companies. These applications are resulting in an exponential increase in the amount of data that utility companies are collecting. Appropriate communication infrastructure and its management is vital for providing this data to unlock the full potential of the SG. Typically, these applications generate different types of data traffic that can be divided into multiple traffic classes with different QoS parameters (priority, throughput, latency etc.). Traditionally, these classes are handled with static network configuration based on individual application policies. However, due to increasing network dynamism, the problem arises as to how to adjust these configurations, based on changing traffic situations. In this paper, a software defined networking (SDN) based solution for distributed and dynamic Smart Grid network management is presented. Proposed solution responsiveness to complex dynamicity of Smart Grid communications is evaluated on a developed evaluation platform for the following cases: (1) Automatic Generation Control (AGC) during peak load, (2) Volt/Var optimization (VVO) during peak load, (3) steady-state operation with static (background) traffic load, (4) stress-state under continuous background traffic overload and (5) dynamic prioritization of traffic for data disaggregation. The presented solution provides significant benefits, when compared with traditional networking in tested scenarios, including: over 70 times lower latency for the most time-sensitive traffic (AGC), 25% increased VVO system observability and 5% to 7% decrease in unprivileged traffic bandwidth consumption whenever privileged traffic QoS is threatened. Additionally, it is shown that dynamic prioritization can provide requested QoS on demand as long as overall capacity is larger than the privileged traffic offered load.

---

1. Introduction

Smart Grid is the next generation power grid. It is expected to be efficient, reliable, easily extendable, secure and able to support the ever increasing number of devices [1] as

---

☆ This document is a collaborative effort.
*Corresponding author

Preprint submitted to Elsevier *November 4, 2018*

well as growing energy demands [2] in the not so distant future. Since the prerequisite for successful Smart Grid implementation and deployment is in bi-directional information flow (i.e. from utility to field devices and customers and vice versa), the existence of appropriate advanced communication infrastructure is essential [3] [4] [5] [6]. Providing quality of service (QoS) for Smart Grid communication traffic, while taking into consideration dynamic re-prioritization, is addressed in this paper.

The Smart Grid communication infrastructure will have to cope with a large number of communication subsystems and be highly adaptive in order to support (growth) trends that are similar to what was observed in the last decade. In the early days, power grid communication systems were used to connect a relatively small number of devices using leased lines or point-to-point radio links [7], often through low rate serial protocols and early SCADA systems. That was followed by the deployment of Power Line (Carrier) Communication technology providing communication mostly through power lines at high voltages with modest increase in data rates. More recently, a number of different technologies are increasingly used in power grid communication subsystems - from cellular, Wi-Fi, Zigbee, broadband Power Line Communication [8], and leased IP links to novel approaches such as Random Phase Multiple Access technology that has already been selected by Riverside Public Utilities for deployment [9]. At the same time, the public internet has reached almost every household in first world countries and has improved regarding quality and bandwidth. The public internet will be increasingly used for data acquisition, since a majority of end-user equipment can be trivially connected to it and deploying and maintaining a dedicated communication network is prohibitively expensive for individual utility companies. In addition, even for the equipment owned by specific utility companies, creating dedicated networks on a large scale to ensure peak response can turn costly. Utilities will rely on the public internet infrastructure for at least some of their future communication needs [10].

Another change that is likely to emerge is the push of both aggregation and fast control (SCADA, phasor measurement unit (PMU), advanced metering infrastructure (AMI), electric vehicles (EV), etc.) much closer to the consumer in order to achieve fine-grained bandwidth utilization and management similarly to how Netflix [11] and Google [12] are pushing their services closer to the network edge.

One of the top research topics in Smart Grid is observability and control [13]. Missing a timely response can have serious consequences [14], with the estimated annual cost of power outages reaching \$150 billion (which is equivalent to a kWh price increase of 4 cents [15]). Additionally, a slow response time of the grid devices is the most common cause of blackouts and burnouts [3]. Also, the grid itself is not controlled by a single entity (i.e. 3500 participants are involved in North American system stability [16]), resulting in increasing complexity of grid observability, analysis and control.

At the same time, the number of devices will only continue to grow, and increasingly these devices will have to be addressed individually to achieve full manageability and cost savings. The Smart Grid communication network can be classified as: Home Area Network (HAN), Neighborhood Area Network (NAH) and Wide Area Network (WAN) consisting of a Backhaul Network and a Core Network [3], [17] with a number of industrial protocols and technologies including: DNP3 [18], IEC 61850, C37.118.1/2 [19], [20], etc. The focus of this work is on IP based networks because: a) utilities currently rely heavily on IP [21] and b) it is expected that IP will become even more dominant [22]. IP based networks typically provide best effort service meaning there is no guarantee that data will be delivered, and whether it will be delivered inside a certain time window. The time needed for message delivery depends heavily on the network load because of traffic multiplexing.

The Smart Grid should support a number of applications, each potentially having specific network requirements with respect to three major parameters: priority, bandwidth and latency. While a number of these applications require real time performance (e.g. SCADA, OMS, DER and PMU [23]) or near-real time like AMI [17], there are a number of cases where real time performance is not needed (e.g. configuration data, data generated by equipment while testing it, or historical data). Regardless, time-aligned data can have have significant impact [24] not just on the performance but also on the way SG applications are implemented (i.e. state estimation data can be received from the field without executing non-linear algorithms typically used for this purpose and with synchronized clocks, devices can execute time lined switching plans [16]). An additional variable that has to be taken into consideration is that these requirements can change for the same data source depending on the application using it [25], i.e. AMI data has lower priority when polled for electricity billing than when used for demand response (DR). From a communication point of view, each of them can be treated as a separate flow with its requirements. Based on this, certain Quality Of Service (QoS) has to be provided [26] and priority of service is crucial [17]. There are multiple approaches in computer networking for fulfilling these requirements: a) bandwidth over-provisioning, b) application level optimization, and c) implementing QoS on a communication infrastructure level.

Using bandwidth over-provisioning, a utility installs or leases maximum bandwidth statically. This approach is expensive especially if there is a high difference between the typical and maximum bandwidth requirements. For example, in SCADA the systems consumed bandwidth varies during the day. From the authors' experience, nightly consumption is typically only half of the bandwidth consumed during peak hours. It should be noted that Smart Grid communication trends are quite similar to Internet trends. This can be easily illustrated with data presented at CAIDA for the Chicago passive network monitor A, where the difference between minimum and maximum bandwidth is 78% [27]. In order to guarantee a reasonable level of service, WAN links are typically provisioned with 30-40% average "utilization" [28]. Additionally, peak network consumption is reached during high filed activity or during critical events with highest impact on QoS fulfillment.

Even with expensive over-provisioning, once bandwidth requirements reach hard limits due to growth, high priority traffic could be jeopardized by lower priority traffic. Taking into consideration floating traffic priority and grid state, ensuring high performance and effective communication flow could be extremely difficult with traditional networking mainly because provisioning is not dynamic and does not scale easily.

New Internet of Things (IoT) deployment with large numbers of sensors (smart meters, environmental sensing, etc.) is increasingly becoming a part of the SG deployment and by definition relies on the transport over the public internet. Also, the number of geographically distributed sub-systems (such as solar or wind farms, EV stations, etc.) that are connected to SG is growing. These sub-systems are increasingly connected over the public internet to the dedicated utility network. Both of these cause traffic interference between power utility traffic and the 3rd party network traffic which is typically not controlled by power utility company.

Aside from QoS, each data flow has its own priority compared with other data flows.

Depending on the situation, this priority can change dynamically [2], with implicit dependency on both application state and the grid status.

All of these factors: natural SG traffic variability based on field events, growth in number of devices due to SG evolution and IoT integration, growth in SG applications and an increase in network heterogeneity, are the significant contributors to communication network traffic variability, resulting in a six-fold increase in network traffic [29]. Static over-provisioning based on this extreme case scenario would result in significant cost increase of dedicated network deployment. Similarly, the QoS only based provisioning lacks situational awareness of the SG. To address these shortcomings, this paper proposes an SDN based solution for the Smart Grid communication infrastructure that is providing dynamic traffic prioritization and support for requested QoS guarantees across a range of typical traffic situations. The proposed SDN controller is tightly integrated with a power distribution management system and is fully aware of deployed SG applications and their requirements. Proposed solution advantages against traditional networking are shown on the execution of AGC and VVO Smart Grid functions, data acquisition during peak and constant load and data dis-aggregation. The evaluations using the proposed solution show that, as long as high priority traffic bandwidth does not exceed available bandwidth, peaks and massive background traffic loads will not have significant impact on critical system performance. Using this approach, it is possible to manage the dynamic nature of Smart Grid communication traffic as well as enable the introduction of new Smart Grid applications at run time without down time, or the need for communication infrastructure reconfiguration. The novel contribution of this paper is in managing the SG communication network based on the dynamism of SG applications by using SDN capable infrastructure.

The remainder of this paper is organized as follows: related work is discussed in Section II, requirements are presented in Section III, proposed solution architecture, design and implementation are covered in Section IV, while the performance results are presented in Section V. Finally, Section VI concludes the paper and describes directions for future work.

2. Related Work

A number of papers using different communication architectures with application level optimization for minimizing bandwidth utilization in Smart Grid communications have been introduced in the last couple of years. One notable solution, based on UDP and decentralized application execution, is presented in [30]. As is the case with a majority of application level approaches, this work, while addressing some of the QoS requirements, doesn't take into account bandwidth increases and dynamism that is due to power system growth, addition of new devices and/or SG applications.

Another set of approaches is based on QoS implementation at the communication infrastructure level. A typical example for this class of approaches is the use of Multiprotocol Label Switching (MPLS) to support QoS by using traffic engineering and divided Smart Grid traffic in four classes [6]. The use of MPLS Traffic Engineering with DS-TE, active queue management algorithms and RIO showed significant traffic delay reduction. One drawback of using MPLS, as stated in [31], is a) the time it takes to reconfigure the network which might be prohibitive for a highly dynamic network and b) that adding new services includes implementing them on each router. When compared to SDN, MPLS

TE suffers from two typical problems: (1) poor efficiency because services send data when they want without taking the network state into consideration and (2) poor sharing since achieving global utilization optimum needs information from the whole network [32]. Also, the proposed solution carries 60% additional data compared to MPLS TE for the inter data center WAN. An additional drawback of MPLS is interoperability - there are no facilities for combining configuration between different ISPs [16], for the case of independent autonomous systems.

SDN is being supported significantly by network providers such as Microsoft [32], Google [28] and Amazon, employing it in their data centers and equipment vendors such as NEC, Juniper and Cisco as stated in [31]. It is being used for centralized network control and monitoring, traffic engineering capable of responding to dynamicity of network requirements in normal and irregular operation modes, and increasing network utilization to avoid over provisioning, scalability and security. There are a number of publications pointing out the benefits of using SDN in power system management [31], [33] including: providing global view and control, software defined network configuration, and bandwidth on-demand. Similarly in [21] the authors show how SDN can be used to fortify Smart Grid communication network resilience. The same work also points out that, without SDN, IP based communication in grid communication networks are in most cases hard-set when the system is designed and, giving routing as an example, re-configuring the network once deployed, can be quite hard. The benefits of using SDN in SG is to streamline network management and simplify the addition of new functionality through controller programmability as shown in [5]. [34] focuses on network resilience by using redundant links and using SDN for link selection in case of link failure. [35] presents an SDN based solution for collecting PMU data. Network bandwidth is saved by filtering generated data based on subscribed party rate requirements. It does not cover prioritizing different traffic types but optimization at the packet routing/switching level. [36] presents industrial internet of things with focus on providing QoS using appropriate routing. Dynamic priorities are not covered in case available bandwidth is insufficient. The approach presented in [37] provides a proposed solution for guaranteeing a deterministic practice for IEC 61850 based networks for substation automation purposes while using static QoS assignments. The authors in [38] developed the SDN4SmartGrids test bed with four switches, one SDN controller, two servers for load generation and one client for traffic receiving. This testbed was used for implementation and performance evaluation of a fast recovery algorithm and showed promising results regarding using SDN for fast recovery. This work also implemented QoS guaranties by using predefined (and static) bandwidth allocation for a number of SG applications.

A comparison between MPLS and OpenFlow was presented in [31] with a conclusion that OpenFlow switches can perform as well as MPLS when deployed in Smart Grid communication networks. Similarly, a Smart Grid communication solution based on implementing MPLS features in OpenFlow proposed in [7] shows that it outperforms MPLS alone.

An SDN based solution for Smart Grid communication infrastructure presented in [16] defines data delivery requirements for a wide area measurement system for data delivery (WAMS-DD), which is designed with the same motivation as the Dynamic Prioritization. It identifies five requirements for data delivery: (1) Hard end-to-end guaranties over the entire grid; (2) a long life-time, future-proof solution; (3) use Multicast as the normal mode of communication; (4) Provide ultra-low latency requirements (8-16ms) over hundreds of miles; (5) Provide extremely high throughput, needed by devices such as synchrophasors and digital fault recorders.

A number of relevant inter-domain aspects of SDN networks are also researched by the community. One example of such research, is the Software Defined Internet Exchange prototype [39] that is successfully scaling to hundreds of participants and policies, and providing a flexible solution for packet routing between independent domains.

3. Requirements

As stated earlier, in this paper we present an SDN based solution for Smart Grid network management supporting QoS and dynamic re-prioritization. A proposal for Distributed Real Time Data Collection and Management System for Smart Grid based on SDN is presented with five usual test cases to prove its usability.

The following requirements were considered in the design:

1. Smart Grid communication traffic can be divided into traffic classes: Traditional software for grid observability and control is usually divided based on data source types such as SCADA, AMI, PMU, Electric Vehicles, Video surveillance, etc. [17] (and as time goes on, this list will only continue to grow). At the same time, a number of other applications, that are not directly related to power grid management, are also consuming networking resources. This paper, assumes that the SG communication traffic can be assigned, without loss of generality, to one of the following classes: (a) SCADA, (b) PMU, (c) AMI, (d) Corporate traffic. The first three belong to SG applications communication traffic that is associated with operation of the power network. In contrast, Corporate traffic includes other utility traffic that is generated by applications but is not received from the field nor sent to field devices or customers. Examples of this are replication data, data sent to client applications, web traffic, data backups, etc.

2. Each communication traffic class has dynamically changing priority: while each traffic class has nominal priority, it is important to emphasize that these can change dynamically [2], [40]. One example of this dynamism is explained in [2] where low priority AMI measurement should be given higher priority and lower delay allowance if they are in the area where DR is executed. Similarly, higher priority is given to PTT or video traffic during an emergency or after an incident has been detected. The need for prioritization during field devices commissioning is another example of a need for changing nominal priority. Misconfigured RTUs sending complete change history whenever there is a connectivity issue can result in significant unnecessary network traffic endangering regular operation. This suggests that any newly installed equipment still in the testing phase should be added as a separate, low priority flow with significantly limited bandwidth allocation. Once a new device is ready for production, it can be promoted to its respectable traffic class.

3. Traffic class bandwidth requirements can change dynamically: Typically, bandwidth requirements depend on a number of parameters. As stated earlier SCADA bandwidth significantly varies by the time of day. Similarly, a number of applications that run intermittently result in highly dynamic traffic. One use-case is

on-demand data disaggregation. Applications of non-intrusive appliance load monitoring are multiple, as stated in [41]: gaining a better understanding of consumer behavior in order to achieve more precise load forecasting, tracking of consumption by devices, better customer participation in decision making and verification of DR execution. Consumer disaggregation can be helpful in the transition period in Smart Grid while not all devices can provide power consumption information by design. It can also be used by law-enforcement systems for surveillance of activities inside a particular housing unit. Disaggregation significance in Smart Grid is thoroughly elaborated in [42] and states that optimal approach for disaggregation information collection is through the AMI.

4. SDN Control Framework Design and Implementation

The communication subsystem model used in this work is based on a typical communication deployment of a power utility consisting of a number of edge-networks that are connected to the utility control center though a backbone network (typically WAN based). An additional level of abstraction can be introduced by observing that each edge-network essentially acts as an autonomous system (AS) (directly following the Internet organization as a collection of a large number of autonomous systems). Each of these AS can be utility owned, leased or even provided by third party companies such as broadband, cellular or telecommunication internet providers. Similarly, an AS could be completely owned by a third-party company to outsource data aggregation. Therefore, a communication network is modelled as directed graph $G = (V, E)$ where V is a set of nodes and E is a set of directed links. Each link is defined by endpoints $l(x, x') \in E$ and capacity $c(x, x')$ while $x \in V$, $x' \in V$. Autonomous systems, including core and utility networks, are disjoint sets of nodes. Each node $x \in V$ belongs to only one autonomous system, meaning $V = \cup S_i$, where S_i represents autonomous system.

The initial modeling assumption is that the core network is under the administrative control of the utility company because the utilities are typically not comfortable with networking infrastructures they do not own [17], claiming: (a) that they need priority access over consumers especially in critical situations (such as natural disasters, bad weather, etc.) [17], and (b) that their own deployment is significantly more resilient since it is based on proprietary systems [17]. The second modeling assumption is that the equipment installed at both in the core and utility premises is SDN capable. This assumption could easily be extended to autonomous systems for the increasing benefit of greater controllability (but is beyond the scope of this work).

The proposed solution has typical SDN, three-layer architecture - application layer, control layer and data layer. The data layer is assumed to be using standard SDN-based forwarding mechanisms to move application generated data traffic through the network. A dynamic set of SG applications are sources and sinks for that traffic and live in the application layer. Finally, the control layer is a logical entity that is used to manage the networking components. In this work, it is assumed that it is managed by the Smart Grid SDN controller (SGSDNC) that, in addition to standard SDN control functions, includes the implementation of the SG decision module (DM).

The SG traffic model is explained as follows. $A = a_1, ..., a_k$ is a list of running privileged SG applications ordered by priority, from the most important to the least important application. Each privileged SG application $a_i \in A$ is specified as $a(b_a, E'_a, p_a)$.

b_a is the bandwidth required for the application, E'_a is a set of links affected by the application and p_a is unique application priority. Available bandwidth per link l is calculated as shown $k_l = c_l - \sum\limits_{a \in A} b_a | l \in E'_a$. K'_a contains available capacities for links in E'_a.

4.1. Dynamic Prioritization Algorithm

Algorithm 1: Update rules on switches if possible.

```
Input:
$G(V, E)$: communication network model;
$A$: Smart Grid applications running, sorted by priority;
$a_m(b_m, E'_m, p_m)$: SG application requesting dynamic priority increase;
Result:
Switch rules are updated according to traffic priority and bandwidth constraints.
Data:
$A_p$: Privileged applications sorted by priority
$A_u$: Unprivileged applications sorted by priority
begin
    $A_p = EmptyList()$
    $A_u = EmptyList()$
    $A.add\_sorted\_by\_priority(a_m)$
    foreach $a \in A$ do
        if $FlowCanBeAddedAsPrivileged(A_p, a) == True$ then
            $A_p.add(a)$
        else
            $A_u.add(a)$
        end
    end
    $DeleteRulesFromSwitches(A)$
    $InstallPrivilegedRule(A_p)$
    $InstallUnprivilegedRule(A_p)$
end
```

Static bandwidth allocation lacks adaptive mechanisms to combat network dynamics [43] and under certain traffic conditions, static provisioning results in violation of delivery deadlines for high priority traffic. Dynamic prioritization is used to ensure that the most important traffic will be favored and the QoS is respected [43]. The core idea of the dynamic prioritization is to dynamically reassign network traffic priorities increasing the importance of high value traffic as the the overall network traffic worsens. Algorithm 1 represents how a request for dynamic prioritization is processed. Changing priorities of current applications and/or adding new privileged application consists of creating a list of applications sorted by priority, and then checking if the network links can withstand the additional load, from the most important to the least important application. If so, the application is added to the list of privileged applications, otherwise it is added to the

Algorithm 2: *FlowCanBeAddedAsPrivileged()*: Decide if request for dynamic priority increase can be executed:

```
Input:
$A'$: privileged Smart Grid applications currently running;
$a_m(b_m, E'_m, p_m)$: SG application requesting dynamic priority increase;
Data: $K'_m$: available capacities for links in $E'_m$.
Result: $P(a_m) = \begin{cases} 0, \exists K'_m, b_m > k \\ 1, \exists K'_m, b_m < k \end{cases}$ : returns 1 if requested capacity is
        available and 0 otherwise.
begin
    foreach $k \in K'_m$ do
        $if(k < b_m)$ return 0
    end
    return 1
end
```

list of unprivileged applications. Algorithm 2 is used to determine if adding a privileged flow is possible.

The resulting set of SDN forwarding rules that is produced by Algorithm 1 is sent to the networking fabric at the end of each run.

4.2. SDN Smart Grid Controller Implementation

OpenFlow with POX controller[1] was chosen as an SDN implementation platform. POX is a modular SDN application development platform written in Python [44].

The controller has the responsibility to receive requests from the SG application layer and, based on decision modules response, installs/modifies/deletes forwarding rules on an SDN network element (switch/router). It consists of the following components:

1. TCP server receives requests from the application layer. It exposes the following interface (through binary protocol):

   ```
   bool AddModifyFlow(app_type,
                      priority,
                      requested_bw,
                      links)
   ```

2. Decision Module (DM), is responsible for calculating if the request for QoS increase can be achieved and implements decision algorithm explained in the previous section.

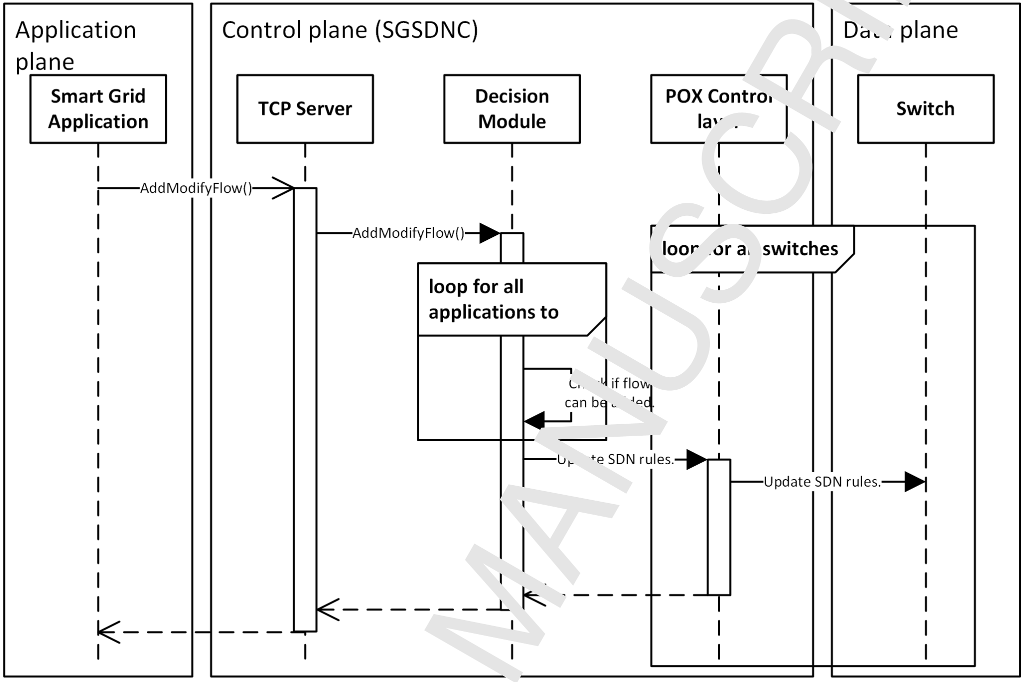
3. OpenFlow (SDN) Control layer is responsible for installing SDN rules at the switches. Control layer uses POX to communicate with switches.

---

[1] Authors considered using OpenDaylight controller but determined to use POX since it is lightweight, simple and well documented.

Interactions between Controller components are shown in Fig. 1.

Figure 1: Interaction between framework components



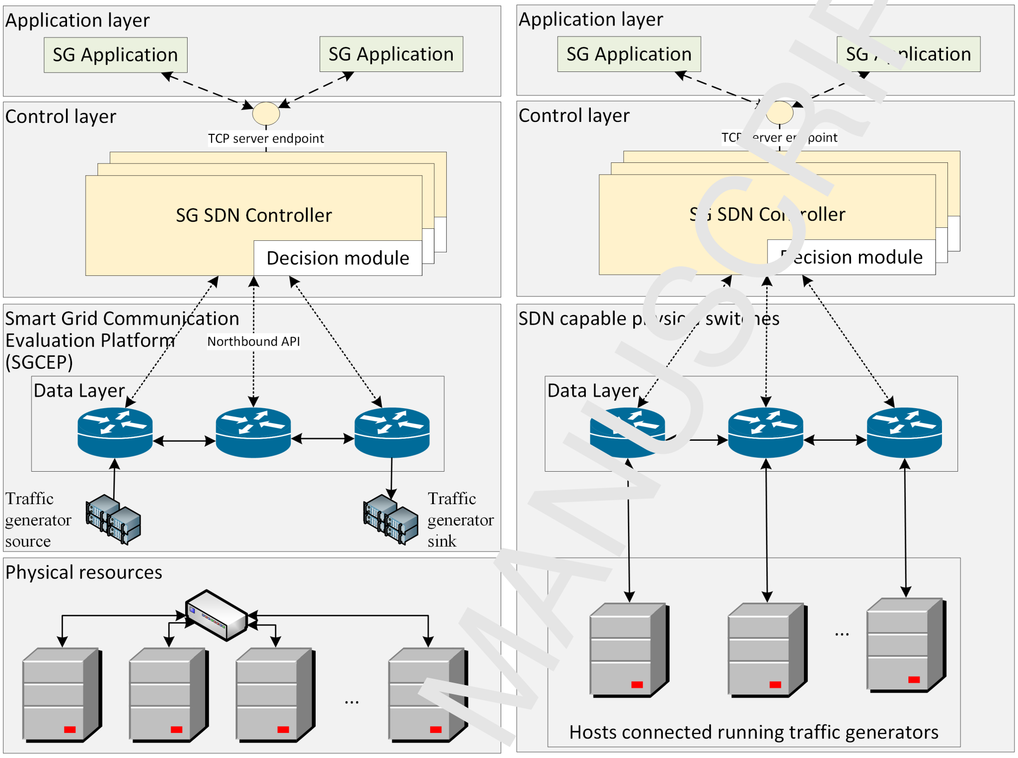
The DM implements both Algorithm 1 and 2 in Python. It has low time complexity, $O(n)$, where n is the number of affected links. The SGSDNC, among other things, receives and processes requests for priority changes, issued by a specific application or other actors in SG. A request consists of: application type, priority, bandwidth requirements and affected links. Based on that information, the DM calculates if all the links can withstand the additional load without degrading currently provided QoS. If so, a rule to treat the requested traffic as privileged is made and installed on the switches by the Controller. The OpenFlow rule is created based on the source and destination IP addresses and ports. The collection of applications is stored in a sorted list, while privileged and unprivileged applications are stored in a pair of sets.

Flow rules are assumed to be updated relatively infrequently, based on the operator decision or the state of the Smart Grid. Therefore, two OFPT_FLOW_MOD messages are sent to all switches for each flow. One to delete the flow (with OFPFC_DELETE command) and another one to add the flow with appropriate priority (with OFPFC_ADD command). Updates to flow tables are assumed to be on the order of 160 bytes per rule resulting in a modest control traffic that does not significantly impact other (data) traffic flows[2].

---

[2] The implementation supports both proactive and reactive flow management.

Figure 2: Evaluation Architecture



a) Network emulated using Smart Grid Communication Evaluation Platform based on Mininet Cluster edition.

b) Using SDN capable physical switches

Controller treats each flow as a separate traffic class and can handle various types of traffic flows. This is how the first requirement from Sec. 3 is addressed. The Controller covers the second requirement by allowing promotion of certain flow as privileged using the *AddModifyFlow* function. And finally, to support the third requirement, each flow-level bandwidth demand can be changed by issuing additional calls to *AddModifyFlow*.

5. Performance Evaluation

In this section, we provide details on the performance evaluation platform and the set of test cases that were used to evaluate DM implementation.

5.1. Evaluation Platform

To verify the solution, the Smart Grid Communication Evaluation Platform (SGCEP) based on Mininet [45] was developed as shown in Fig. 2(a). Mininet[3] is a network

---

[3] Aside from Mininet, authors considered fs-sdn [46] simulator for performance evaluation. fs-sdn provides resource light network simulation suitable for large networks and simulation faster than real

simulation platform written in Python that uses OS-level virtualization and provides a high level emulation environment. It uses Linux native networking stack and allows applications to execute on simulated hosts. Since executed test cases exceeded processing power of a single computer, they were run on a Mininet Cluster edition that supports running simulations on multiple computer nodes.

The SGCEP has the role to: automate node configuration, configure a Mininet environment based on requested SG topology and data flows, execute the evaluation test and collect results. The size of the test case, as deployed by the SGCEP, is typically constrained with the hardware resources (nodes in the underlying computing cluster and links between them). However, the solution can be tested on any set of appropriate Linux boxes - such as dedicated high-performance cluster hardware or VMs provisioned by a cloud provider. The evaluation platform consists of:

1. **Environment Configuration**: Emulation is executed on multiple computer nodes requiring the environment to be configured on all of them. This is done by a set of scripts receiving a file with a list of nodes the test will execute on, and includes the following:
   (a) Exchanging public keys between nodes to enable SSH log-in without passwords. This is needed since Mininet Cluster relies on SSH tunneling between nodes in a cluster.
   (b) Distribution of executable and configuration files on all servers on the cluster to ensure that all servers are executing up to date code.
   (c) Start network monitoring using NetFlow. This includes starting the NetFlow capture process, nfdump, on each server to capture traffic-specific statistics NetFlow provides. NetFlow is a network protocol released by Cisco for collecting IP traffic information and is used by the SGCEP as primary performance measurement source. It can be used to determine the traffic source and destination, number of packages, bytes transferred and similar information per data flow. It consists of the following components: (i) The NetFlow data source which can be a switch or router. (ii) The NetFlow collector, a node that saves the NetFlow data it received from the data source. (iii) Application for data analysis.
2. **Test Execution**: Smart Grid communication topology needs to be implemented in Python describing the topology and data flows. Smart Grid Communication Network topology and experiment description is provided as an input file to the SGCEP initialization Python code; it executes the following phases:
   (a) Creation of Mininet Cluster topology based on the described Smart Grid communication topology.
   (b) Configuration of link bandwidths of the Open vSwitches [47]. Link bandwidth was limited with traffic policing for ingress traffic, while queuing is used for egress traffic because Mininet Cluster edition does not support Linux Traffic Control.

---

time [46]. It hooks its own functions instead of actual OS specific networking APIs. However, Mininet based solution was chosen mainly because it is much more widely accepted in the research community.

  (c) Creation of Open vSwitch queues used for QoS provisioning. Open vSwitch supports creating queues with minimum and maximum traffic rate. Queues for each priority class are created for limiting minimum and maximum traffic rates. The controller is responsible for installing appropriate rules for directing data to the appropriate queue based on class type and its current priority. Flow information was obtained using NetFlow, a feature supported by Open vSwitch with sampling frequency of one second.
  (d) Starting traffic generators on appropriate nodes.

3. **Traffic Generators**: To test scenarios of interest, appropriate network C programming language based Traffic Generator (TG) was developed. Traffic generators are used to measure the Controller performance. There are two types of traffic generators - (a) stream and (b) command traffic generators. Both have generator (data source) and sink (data destination). Stream traffic generators are meant to generate traffic at a certain rate (i.e. 100KBps) per time slice. This traffic generator was used for SG applications communication traffic when generating at a constant rate or corporate traffic at a constant or variable rate, depending on the test case. A command traffic generator issues commands in bulk and receives responses from the simulated field while measuring round trip time. Depending on the test case, the command traffic sink will send a response after a certain time interval to simulate command execution. Both types of generators receive all information needed for execution as command line arguments (such as sink port, drain port, drain IP address, bandwidth to consume or command specification depending on generator type). The TGs are configured at run time (i.e. instructed to generate stream flow at 100KBps rate) through command line arguments while the communication between generators is based on a client-server model (and implemented through BSD sockets). The SGCEP initialization script is also in charge of starting TGs.

4. **Results Collector (RC)**: A set of scripts for data collection with a primary function to stop NetFlow collection on individual nodes, convert binary NetFlow data on each node and copying them and traffic generator logs from all nodes to a local machine as archive file for further analysis.

The test application, as a representative dynamic Smart Grid Application, was developed to connect to the controller TCP endpoint and send requests for adding the application and/or changing priority of the existing application (i.e. for the AMI traffic priority increase/decrease).

The SGCEP allows for tests to be executed multiple times and results to be collected from multiple nodes for analysis. Using traffic generators it is possible to emulate needed communication flows with different load functions thus creating environments to test the SGSDNC for all three requirements. Lastly, the SGSDNC behavior can be verified by looking at the NetFlow data or traffic generator logs.

It should be noted that it is possible to use the same SGSDNC directly with the SDN capable physical switches without the SGCEP as shown in Fig. 2(b) and that the implemented control plane supports multiple controllers. However, in this paper only results for a case with a single centralized controller are reported.

Smart Grid communication topology (collection of switches, data flows and links) is static, meaning that it is not possible to change the topology during the run time of the test.

Source code for the SGCEP and SGSDNC is publicly available at [48][4].

5.2. Results and Discussion

The SGCEP was executed on the ORBIT testbed [49] at Rutgers University. A network test topology was deployed in the Mininet [45] emulation environment with the Mininet Cluster edition (shipped with Mininet 2.2.1) running on the Ubuntu 14.04 OS. Each entity (traffic generator, switch and controller) was run on a separate physical machine (quad core Intel i7 class CPU, Gigabit Ethernet and 8 to 16 GB RAM) with a resulting evaluation cluster consisting of 34 machines.

Based on [17], it can be concluded that a large portion of data collected in the (future) Smart Grid is not needed for imminent grid control. This means that only traffic with the highest priority has to be favored (regarding network bandwidth and/or latency) while lower priority data will be sent to a control center at a pace consistent with available capacity. Resulting Smart Grid dynamics were evaluated with respect to two parameters:

1. Data dynamics: Smart Grid data volume can depend on many factors such as time of day, day of week or month (working days vs weekends vs holidays), weather and similar environmental factors.

2. Limited network capacity: Over provisioning is avoided and network bandwidth consumed is maximized while providing critical application data needed for execution.

Based on the evaluation parameters specified in Section 3, the traffic was divided into four classes, each belonging to a specific priority group. The traffic for each group was generated by a number of TGs with the following patterns:

1. SCADA traffic at a rate of 1 KBps for each transformer area, based on the experience of the authors.

2. PMU traffic estimated at 10 KBps per transformer area, which was based on reported values in [50].

3. AMI traffic rate calculated using the formula presented in [51], where λ is the number of bytes per second, Z is the number of smart meters, P is the size of smart meter packet and t is the rate of data generation represented in minutes:

$$\lambda = \frac{Z \times P}{60 \times t} \tag{1}$$

This formula assumes that smart meters will not simultaneously (i.e. synchronously) send data but will rather have uniformly distributed packet transmission start times. As stated in [51], packet size P is 512 bytes in current standards and the

---

[4] Tutorial for the environment setup, test execution and collecting results is available at the same location. Results presented in this paper can be reproduced using the source code available.

data generation period t is 5, 15, 30 or 60 min. In this work, for test cases 1-4, a 5 min generation period was used. For the last test case 15s was used for a part of the network. Table 1 sums kilobytes per second for different number of meters depending on data generation period, used for testing purposes.

Number of customers	Generation period [min]			
	5	15	30	60
5000	8.3	2.8	1.4	0.7
10000	16.7	5.5	2.8	1.4
20000	33.2	11.1	5.5	2.78
40000	66.7	22.1	11.1	5.5

Table 1: AMI bandwidth in KBps depending on number of customers and data generation period

4. Corporate traffic, with different bandwidth footprint, depending on simulation case.

Since network topology was based on a couple of municipalities in Serbia, the number of customers was estimated based on the number of households [52], while the number of transformers was calculated under the assumption of 10.000 customers per transformer area (typical numbers for SCADA for power system and ADMS). The resulting AS assignment is shown in Table 2.

City name	Customers	Transformer areas	Autonomous systems
Novi Sad	120.000	12	4
Zrenjanin	45.000	4	4
Sombor	35.000	6	2
Sremska Mitrovica	27.000	4	2

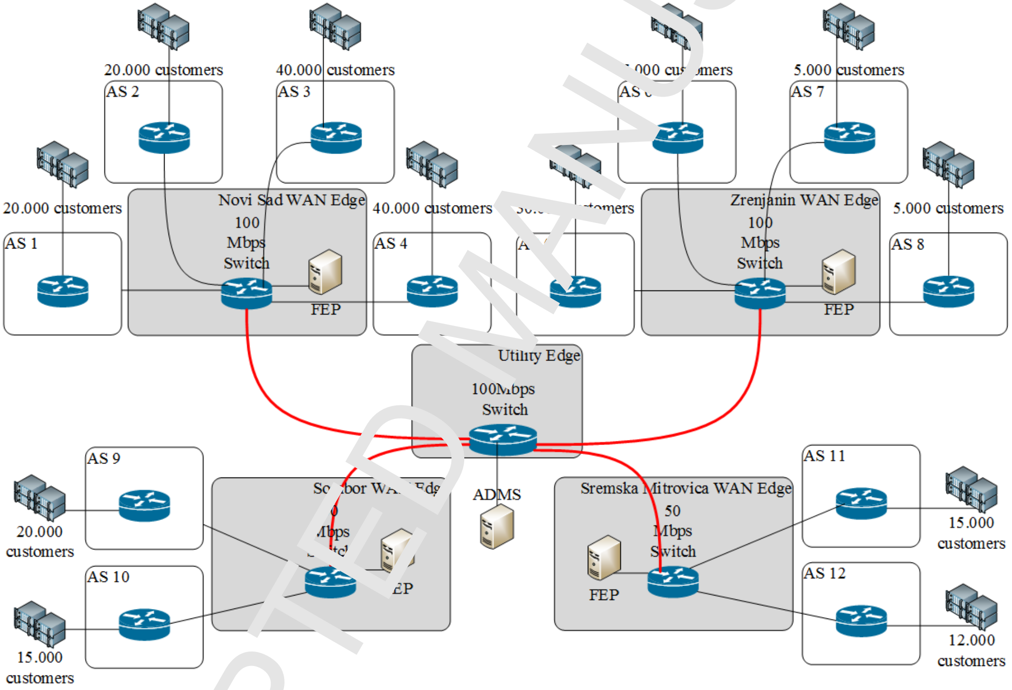
Table 2: Estimated number of customers per area

Based on the data in Table 2 and assumptions regarding the Smart Grid communication network, the test topology is shown in Fig. 3. There are 12 autonomous systems that cover four geographical areas of four cities: Novi Sad, Zrenjanin, Sombor and Sremska Mitrovica.

The following typical test cases are simulated:

1. Measuring AGC command execution delay with constant corporate traffic load.
2. Measuring the duration of executed commands and command execution time for Volt/Var optimization with constant corporate traffic load.
3. Measuring the variable corporate traffic load influence on SG applications communication traffic.
4. Measuring the constant corporate traffic load influence on SG applications communication traffic.
5. Data disaggregation using dynamic prioritization.

Figure 3: Test network topology



Measuring the maximum AMI aggregation frequency is based on the available bandwidth using dynamic priorities to provide data for disaggregation. Generated bandwidth for a complete communication network based on model and test cases shown above is summed in Table 3. Note that AMI traffic for 15s generation period is provided only for AS11 and AS12.

SCADA	PMU	AMI	
		5min generation period	15sec generation period
212Mbps	2.12Mbps	3.1Mbps	7.4

Table 3: Generated traffic

Each test scenario was run with the two SDN controller types i.e. with and without dynamic traffic prioritization support.

5.2.1. Test Case 1: Automatic Generation Control during maximum load

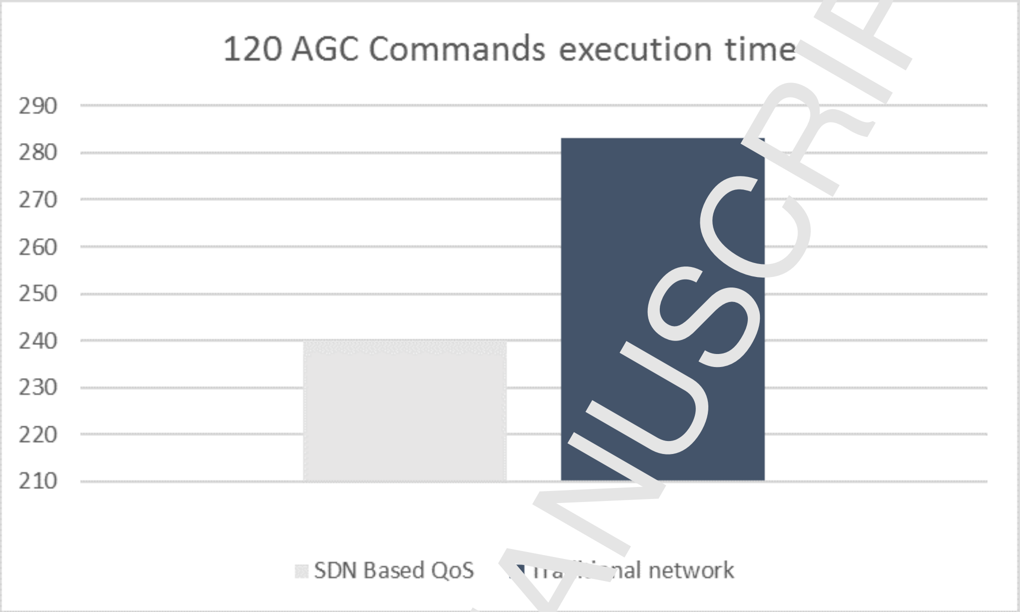
When imbalance between power consumption and production occurs in a power system, the frequency drops or raises, e.g. when production decreases because one or more generators trip, or the load in system increases. If a system can absorb this change, it is called self-regulation of consumption by frequency. If imbalance between production and consumption cannot be self-regulated - it is necessary to issue control commands to generators by increasing or decreasing frequency in order to return frequency to normal value to minimize production costs and operate the system at an adequate level of security [53]. The role of Automatic Generation Control (AGC) is to automate this process. It can be executed on a single (isolated) area or multiple connected areas [53]. When AGC is executed on multiple connected areas, it is mandatory to execute AGC simultaneously on all areas [53]. As stated in the same paper, a typical period for data acquisition and the decision cycle is 2 or 4 seconds. This means that keeping communication performance needed for data acquisition and control is crucial for successful AGC.

This test case presents a simulation of AGC executed on multiple connected areas from the communication point of view by sending 120 subsequent commands within a period of 2 seconds. It was estimated that the command execution time was 1.8s. The test was executed with traditional network and SDN based QoS during a maximum background load.

Therefore, for this test case to succeed, it is needed to treat AGC commands and their responses as privileged traffic, thus allowing commands to execute with the period of 2 seconds to provide prompt response to the power system. 120 commands execution should finish in 240 seconds.

Fig. 4 compares the duration needed to execute 120 AGC commands when using SDN Based QoS and traditional networking. Under a maximum background load, while using SDN based QoS, all 120 commands are executed under an expected period of 240 seconds, respecting the command period of 2 seconds. When using traditional networking for AGC under the same conditions - time constraints are not met and 283 seconds are needed to execute all the AGC commands. A delay of 43 seconds (around 358ms per command) is introduced. Since a delay is introduced per command, it will be accumulated during time. If we take into consideration that delay can vary in different parts of the network (i.e. autonomous systems), commands will reach late and out of sync between different generators

Figure 4: Execution time for 120 AGC commands



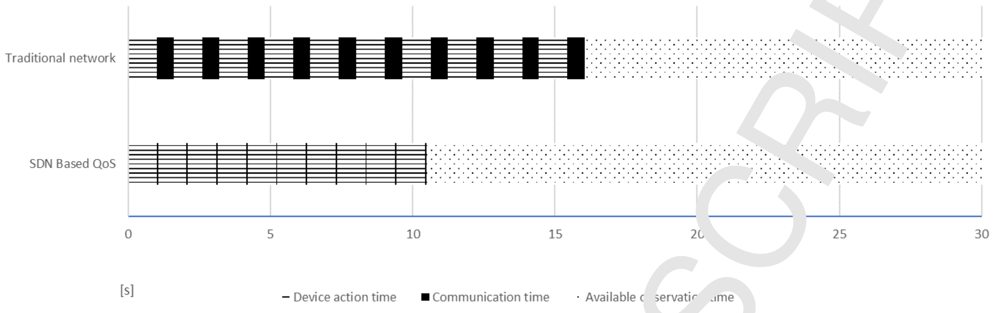
If generators are receiving outdated commands, the system will never reach an optimal state because commands are sent based on system state that is not up to date. Depending on the power system state - it can lead to system outage. It is shown that AGC communication time and latency can be kept as low as possible using the proposed solution without influence of background traffic load.

5.2.2. Test Case 2: Volt/VAR execution during maximum load

This test scenario provides simulation of the Volt/VAR Optimization (VVO) execution. VVO is a Smart Grid application for decreasing losses and increasing grid efficiency [54]. It is one of the core applications used for control in a power distribution network operation. It can be said that this is one of the Smart Grid critical applications because any failure in it's operation can cause load shedding (thus causing planned outage and leaving certain customers without electricity). If a load shedding application is not available, it could even lead to an unplanned outage or in the extreme case to a full system blackout. SCADA provides field data to the ADMS VVO module, which uses it for calculations and provides SCADA with the switching sequence that needs to be executed on the field. The switching sequence consists of commands that should be executed on one transformer area and then the system is observed for effects of commanding until the current control period passes.

As stated in [55], control frequency varies from 1 to 15 minutes and during this period the transformer area usually receives a series of subsequent commands. In this simulation case, control frequency is scaled down to 30 seconds, issuing a series of bulk commands per TS therefore simulating full grid VVO. Command response time was set to 1 second

Figure 5: VVO test results



(i.e. the time interval between the mechanical actuator receiving the command and reporting execution back to the issuer was 1 second). The number of commands per each sequence is 10 consequent commands. The network was continuously loaded as much as possible with corporate traffic.

It is important to execute planned commands as soon as possible in order to maximize the system observation time (until next commanding iteration). The irreducible part of command execution is the time the device needs to execute the command itself, but the communication time of sending the command and receiving the response needs to be kept as low as possible.

Fig. 5 presents the comparison of a single VVO switching sequence under maximum load while using SDN based QoS and traditional networking. Typical results are shown for one transformer area covered with AS1 (other areas are having similar results and were thus omitted for lack of space). The most important observation from these results is that there is an order of magnitude difference in VVO command communication time reduction as well as a significant latency reduction in VVO command delivery to the actuators. The accumulation of increased communication times is seen on the figure as well, based on the available observation time. Test results show that when using a traditional, IP-based, network there is 14.9s available for system observation while when using the proposed, SDN based, approach observation time is 20.5s resulting in increase of more than 25%.

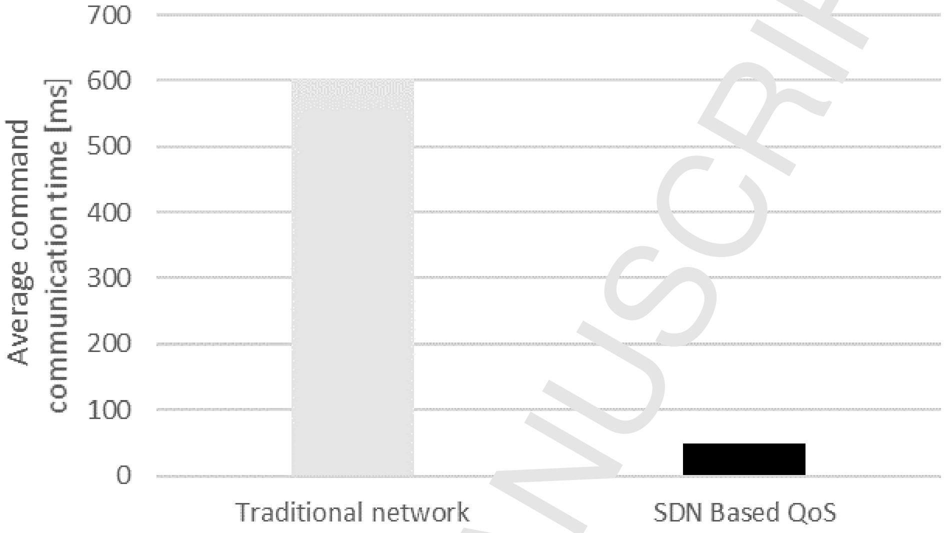
Fig. 6 presents average communication time per command when using traditional network and SDN based QoS. When using traditional network it is 603.43ms while when using SDN based QoS, it is 47.92ms. Using traditional, IP based, networking has shown 1200% longer average communication time than when using SDN based QoS.

Not being able to successfully execute VVO would (potentially) lead to devastating consequences for the distribution network as mentioned earlier. It should be emphasized that when using SDN based QoS, VVO commanding frequency can be decreased if needed as much as the communication network can respond in ideal conditions.

5.2.3. Test case 3: Load influence on data acquisition

This test case covers a usual operation mode when the variable corporate load reaches 10x the maximum available capacity at utility edge. This test case demonstrates how the

Figure 6: Average command communication time



load management behaves with peak load and how it affects Smart Grid communications.

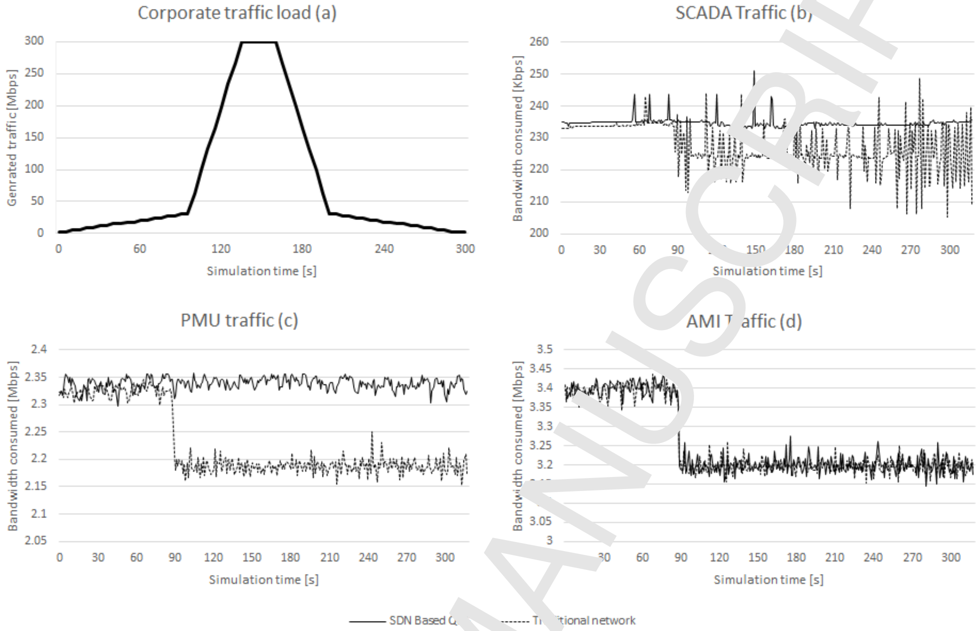
An increase in background traffic over the maximum available bandwidth must not impact the guaranteed QoS for the privileged, SCADA and PMU traffic.

The traffic was generated as shown in Fig. 7 (a). The load was increasing for the first 135 seconds until it reached 30 Mbps, followed by the maximum intensity load for 25 seconds. Then it decreased down to 3% of available AS bandwidth at the end of the experiment.

Corresponding SCADA traffic bandwidth consumption is shown in Fig. 7 (b). It is scaled to provide better insight on the influence of the load depending if QoS is applied. Dotted lines represent traffic when using traditional network, while traffic when using SDN based QoS, is represented with full line. For traditional, IP-based networking, once the link is saturated with the background traffic - the SCADA bandwidth decreases, while it is maintained for the SDN based network. Fig. 7 (c) and Fig. 7 (d) are showing PMU and AMI traffic. It can be observed that AMI traffic QoS was not preserved since it is not treated as privileged traffic compared to background load traffic while PMU traffic has maintained expected QoS in the SDN case. Such a drop in bandwidth will significantly decrease latency.

This test shows that during peak load, a traditional approach results in a bandwidth decrease ranging from 5% to 7% percent while using the SDN approach shows no drop in throughput for the privileged traffic. A peak in communication could happen when there is an increase in events taking place on the grid or it could be related to certain business actions. Either way, during this period, information needed for critical actions could arrive too late if guaranteed throughput is not met.

Figure 7: Load function and bandwidth consumption



5.2.4. Test Case 4: Maximum load influence on data acquisition

This test shows the impact of a sudden and continuous high load of corporate traffic on privileged traffic which is typical during maintenance (e.g. network based backup) or certain administrative actions (e.g. inventory, payroll periods, etc.). If the these peak corporate loads last for a significant period of time, they have potential to cause prolonged ADMS control outages.

Privileged traffic must not be influenced by background traffic to ensure guaranteed QoS for privileged SG applications.

Results are presented in Fig. 8. Background traffic was generated with maximum load as shown in Fig. 8 (a). Fig. 8 (b) shows comparison of SCADA traffic when using traditional networking and SDN based QoS. PMU traffic bandwidth consumption when using proposed solution and traditional networking is presented in Fig. 8 (c) while AMI traffic is shown in Fig. 8 (d). For the proposed architecture, the best-effort class (i.e. AMI) is influenced by the network load while privileged traffic (SCADA and PMU) is not. In the case of traditional, IP-based networks, the SCADA traffic is reduced by 5% while the PMU throughput is reduced by 6%. Open vSwitch does not decrease flow to zero. This test shows that with SDN based network control, bandwidth can be utilized at maximum load continuously without jeopardizing critical applications.

5.2.5. Test Case 5: Data disaggregation

This test case shows how disaggregation of a customers load profiles can be made possible by using dynamic prioritization. Usage of data disaggregation is explained in

Figure 8: Load function and bandwidth consumption

chapter III. The sampling rate was increased from 5 minutes to 15 seconds, for areas covered by AS11 and AS12, and AMI traffic priority was increased above load traffic. It was assumed that increasing the sample rate for disaggregation would linearly increase the AMI bandwidth requirements.

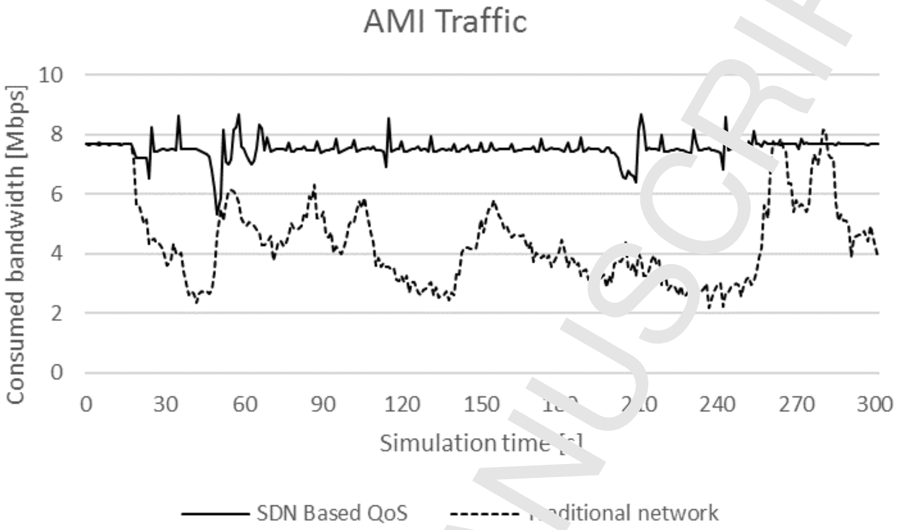
For data disaggregation it is needed to receive data as soon as possible. While AMI traffic is not usually privileged, issuing a request for the traffic priority increase should succeed as long as there is enough resources.

Constant load pressure was applied and targeted QoS was achieved. Fig. 9 shows AMI traffic comparison with and without a proposed solution. When using SDN based QoS, consumed bandwidth by AMI traffic is maintained, it is not fluctuating, as seen when traditional networking is used.

It is possible to increase the sampling rate on any part of the network as long as it can be identified as a separate flow and there is enough bandwidth available for all the privileged traffic. The capability of smoothing traffic performance suggests that employing SDN per AS control can further improve performance of networking in the power system.

It can be noted that when using traditional, IP-based network, bandwidth is fluctuating. This is a consequence of a congestion avoidance algorithm and was verified during this implementation. This behavior is observed and discussed thoroughly in [56]. The congestion avoidance algorithm used during testing is Ubuntu default Cubic. When using SDN based QoS this is not happening because the network is not congested for privileged flows.

Figure 9: AMI traffic bandwidth consumption



6. Conclusion and Future Work

This paper describes an SDN based architecture for providing QoS for Smart Grid network management and control. It explains why an efficient, controllable and extendable communication infrastructure for Smart Grid is necessary. Smart Grid traffic is decomposed into classes, assigning to each of them an appropriate priority. With test scenarios executed it is proven that the presented approach for Smart Grid network management based on SDN does meet requirements and addresses the issues with which traditional IP network configurations have problems and provides dynamic prioritization of network traffic based on Smart Grid application execution. The most important benefit of the proposed dynamic management of the SDN based Smart Grid communication network, is in meeting strict AGC timing requirements and reducing the VVO response time - thus ultimately preventing a possible power system blackout. The proposed solution achieves over 70 times lower latency for AGC commands, and increases VVO system observability by 25%. In addition, it shows significant overall system performance improvement regarding satisfying individual critical application (SCADA, PMU, data disaggregation) bandwidth guarantees by reducing unprivileged traffic bandwidth consumption from 5% to 7%.

With respect to requirements for a data delivery system for Smart Grid applications (as defined in [16]) the proposed solution addresses:

1. Hard, end to end guarantees: covered with dynamicity of priorities and configurable QoS.

2. Lifetime: since the solution is based on IP and SDN, it is virtually guaranteed to have long life time.

3. Low latency: by dynamically controlling traffic prioritization, ultra-low latencies can be achieved (in the case of dedicated networks, latency is fully controllable but highly unlikely to be achieved on the scale of the entire Smart Grid)

4. High bandwidth utilization: ensuring high throughput for critical applications results in high bandwidth utilization (nearing 100% in a number of cases) while respecting required QoS.

Furthermore, each of the areas covered by one autonomous system can be looked at as a micro grid, or a group of micro grids (instead of as a transformer area) which enables expansion to experimentation with a collection of micro grids. Further work will also include development of a Smart Grid Front End Processor (SG FEP) module and the creation of a smart grid aggregation test environment for measuring function execution time and system response time on a significantly larger scale while using industrial protocols such as IEC 61850 and DNP3.

7. References

[1] GTM Research, Trends in Utility Smart Grid Communications Management, Tech. rep., Wood Mackenzie (2013).

[2] J. Spiess, Y. T. Joens, R. Dragnea, P. Spencer, Communication network architecture and design principles for smart grids, Bell Labs Technical Journal 18 (4) (2014) 3–17. `doi:10.1002/bltj`.

[3] J. Gao, Y. Xiao, J. Liu, W. Liang, C. L. P. Chen, A survey of communication/networking in Smart Grids, Future Generation Computer Systems 28 (2) (2012) 391–404. `doi:10.1016/j.future.2011.04.014`.

[4] X. Lu, W. Wang, J. Ma, An empirical study of communication infrastructures towards the smart grid: Design, implementation, and evaluation, IEEE Transactions on Smart Grid 4 (1) (2013) 170–183. `doi:10.1109/TSG.2012.2225453`.

[5] Z. Jianchao, S. Boon-Chong, L. Tek-Tjing, F. Chuan Heng, Opportunities for software-defined networking in smart grid, ICICS 2013 - Conference Guide of the 9th International Conference on Information, Communications and Signal Processing`doi:10.1109/ICICS.2013.6782793`.

[6] M. Yaghmaee, Quality of service guarantee in smart grid infrastructure communication using traffic classification, IET Conference Proceedings (2013) 0803–0803(1).

[7] A. Patil, M. Z. Yunus, S. [illegible], V. Sharma, C. Dickey, Openflow with MPLS in Smart Grids, University of Colorado at Boulder (2014) 1–27.

[8] A. Usman, S. H. Shami, Evolution of communication technologies for smart grid applications, Renewable and Sustainable Energy Reviews 19 (2013) 191–199. `doi:10.1016/j.rser.2012.11.002`.

[9] Ingenu, Riverside Public Utilities Selects Ingenu's Machine Network for Smart City Initiative, accessed: 2017-01-01.
URL `http://www.ingenu.com/2016/04/riverside-public-utilities-selects-ingenus-machine-network-for-smart-city-initiative/`

[10] O. Al-khatib, S. [illegible], W. Hardjawana, B. Vucetic, Traffic Modeling and Optimization in Public and Private Wireless Access Networks for Smart Grids, IEEE Transactions on Smart Grid 5 (4) (2014) 1949–1960. `doi:10.1109/TSG.2014.2304495`.

[11] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, Z.-l. Zhang, Unreeling Netflix : Understanding and Improving Multi-CDN Movie Delivery, Infocom (2012) 1620–1628`doi:10.1109/INFCOM.2012.6195531`.

[12] Google, Google Cloud CDN, accessed: 2018-09-17.
URL `https://cloud.google.com/cdn/`

[13] SmartGrids.eu, Summary of Priorities for SmartGrids Research Topics (June 2013) (2013) 1–10.

[14] A. Bernstein, D. Bienstock, D. Hay..., Power grid vulnerability to geographically correlated failure analysis and control implications, Infocom (Dc). `arXiv:arXiv:1206.1099v1`.

[15] D. E. [illegible], D. J. Phelan, L. Coleman, Voices of Experience: Insights into Advanced Distribution Management Systems, Tech. Rep. 159 (2015).
URL `http://energy.gov/oe/downloads/voices-experience-insights-advanced-distribution-management-systems-february-2015`

[16] D. E. Bakken, A. Bose, C. H. Hauser, D. E. Whitehead, G. C. Zweigle, Smart generation and transmission with coherent, real-time data, Proceedings of the IEEE 99 (6) (2011) 928–951. `doi:10.1109/JPROC.2011.2116110`.

[17] U.S. Department of Energy, Communications Requirements of Smart Grid Technologies, Available online at http://energy.gov (2010) 1–69.
URL `http://energy.gov/gc/downloads/communications-requirements-smart-grid-technologies`

[18] IEEE, IEEE Standard for Electric Power Systems Communications Distributed Network Protocol (DNP3), Vol. 2012, IEEE, 2012. `doi:10.1109/IEEESTD.2012.6327578`.

[19] IEEE Standards Association, C37.118.1-2011 - IEEE Standard for Synchrophasor Measurements for Power Systems, Vol. 2011, IEEE, 2011. `doi:10.1109/ieeestd.2011.6111219`.

[20] IEEE Standards Association, C37.118.2-2011 - IEEE Standard for Synchrophasor Data Transfer for Power Systems 2011 (December) (2011) 1–53. `doi:10.1109/IEEESTD.2011.6111222`.

[21] X. Dong, H. Lin, R. Tan, Software-Defined Networking for Smart Grid Resilience : Opportunities and Challenges [Position Paper] (2015) 61–68`doi:10.1145/2732198.2732203`.

[22] Nist, N. S. Publication, National Institute of Standards and Technology, NIST Special Publication 1108 NIST Framework and Roadmap for Smart Grid Interoperability Standards, Nist Special Publication 0 (2010) 1–90. `doi:10.6028/NIST.SP.1108r3`.
URL `http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf`

[23] W. Wang, Y. Xu, M. Khanna, A survey on the communication architectures in smart grid, Computer Networks 55 (15) (2011) 3604–3629. `doi:10.1016/j.comnet.2011.07.010`.

[24] D. J. Dolezilek, S. Schweitzer, S. E. Laboratories, Practical Applications of Smart Grid Technologies, 11th Annual Western Power Delivery Automation Conference.

[25] J. Taft, P. D. Martini, L. von Prellwitz, Utility Data Management & Intelligence, Cisco (2012) 1–14.

[26] A. V. Akella, K. Xiong, Quality of Service (QoS)-Guaranteed Network Resource Allocation via Software Defined Networking (SDN), 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing (2014) 7–13`doi:10.1109/DASC.2014.11`.

[27] The Cooperative Association for Internet Data Analysis (CAIDA), Passive Network Monitors, Chicago Monitor A, From Thursday 18:00:00 24.03.2016, to Friday 18:00:00 25.03.2016. (2016).

[28] B4: Experience with a Globally-Deployed Software Defined WAN, Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM - SIGCOMM '13 (2013) 3–14`doi:10.1145/2486001.2486019`.

[29] Analyzing iot device performance, `https://www.ericsson.com/en/mobility-report/analyzing-iot-device-performance`, accessed: 2016-09-17.

[30] Y. Wang, P. Yemula, A. Bose, Decentralized communication and control systems for power system operation, IEEE Transactions on Smart Grid 6 (2) (2015) 885–893. `doi:10.1109/TSG.2014.2363192`.

[31] A. Sydney, J. Nutaro, C. Scoglio, D. Gruenbacher, N. Schulz, Simulative comparison of multiprotocol label switching and OpenFlow network technologies for transmission operations, IEEE Transactions on Smart Grid 4 (2) (2013) 763–770. `doi:10.1109/TSG.2012.2227516`.

[32] C.-y. H. Uiuc, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, R. Wattenhofer, Achieving High Utilization with Software-Driven WAN, ACM SIGCOMM Computer Communication Review 43 (4) (2013) 15–26. `doi:978-1-4503-2056-6/13/08`.

[33] Fujitsu, Software-Defined Networking for the Utilities and Energy Sector.

[34] X. Zhang, K. Wei, L. Guo, W. Hou, J. Wu, SDN-based Resilience Solutions for Smart Grids, 2016 1st International Conference on Software Networking, ICSN 2016 (2016) 0–4`doi:10.1109/ICSN.2016.7501931`.

[35] A. Goodney, S. Kumar, A. Ravi, Y. H. Cho, Efficient PMU networking with software defined networks, 2013 IEEE International Conference on Smart Grid Communications, SmartGridComm 2013 (2013) 378–383`doi:10.1109/SmartGridComm.2013.6687987`.

[36] S. Al-Rubaye, E. Kadhum, Q. Ni, A. Anpalagan, Industrial Internet of Things Driven by SDN Platform for Smart Grid Resiliency, IEEE Internet of Things Journal 4662 (c) (2017) 1–1. `doi:10.1109/JIOT.2017.2734903`.

[37] 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`.

[38] N. Dorsch, F. Kurtz, H. Georg, C. Hagerling, C. Wietfeld, Software-defined networking for Smart Grid communications: Applications, challenges and advantages, 2014 IEEE Interna-

tional Conference on Smart Grid Communications (SmartGridComm) (2014) 422–427doi:10.1109/SmartGridComm.2014.7007683.

[39] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, E. Katz-Bassett, G. Tech, U. C. Berkeley, SDX : A Software Defined Internet Exchange, Proceedings of the 2014 ACM conference on SIGCOMM (2014) 579–580doi:10.1145/2619239.2626300.

[40] M. Tahir, S. K. Mazumder, S. Member, Dynamic-Priority based Event Driven Communication in Smart Grid (2014) 1–8.

[41] G. W. Hart, Nonintrusive Appliance Load Monitoring, Proceedings of the IEEE 80 (12) (1992) 1870–1891. doi:10.1109/5.192069.

[42] K. Carrie Armel, A. Gupta, G. Shrimali, A. Albert, Is disaggregation the holy grail of energy efficiency? The case of electricity, Energy Policy 52 (2013) 213–234. doi:10.1016/j.enpol.2012.08.062.
URL http://dx.doi.org/10.1016/j.enpol.2012.08.062

[43] V. Stanisic, M. Devetsikiotis, Dynamic utility-based bandwidth allocation policies: the case of overloaded network, in: 2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577), Vol. 4, 2004, pp. 1958–1962 Vol.4. doi:10.1109/ICC.2004.1312863.

[44] pox, The POX network software platform, accessed: 2018-10-29.
URL https://github.com/noxrepo/pox

[45] B. Lantz, B. Heller, N. McKeown, A network in a laptop: rapid prototyping for software-defined networks, … Workshop on Hot Topics in Networks (2010) 1–6doi:10.1145/1868447.1868466.

[46] M. Gupta, J. Sommers, P. Barford, Fast, accurate simulation for SDN prototyping, Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking - HotSDN '13 (2013) 31doi:10.1145/2491185.2491202.

[47] Linux Fondation, Open vSwitch, accessed: 2016-01-15 (2016).
URL http://openvswitch.org

[48] M. Cokic, I. Seskar, SmartGridNetworkManagement (2018).
URL http://www.orbit-lab.org/wiki/Tutorials/qNSE{#}SmartGridNetworkManagement

[49] M. Ott, I. Seskar, R. Siraccusa, M. Singh, ORBIT testbed software architecture: Supporting experiments as a service, Proceddings - First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Tridentcom 2005 2005 (2005) 136–145. doi:10.1109/TRIDNT.2005.27.

[50] Schweitzer Engineering Laboratories, Synchrophasor FAQs (2012).

[51] B. Karimi, V. Namboodiri, On the capacity of a wireless backhaul for the distribution level of the smart grid, IEEE Systems Journal 8 (2) (2014) 521–532. doi:10.1109/JSYST.2013.2260701.

[52] Households according to number of members by settlements, Serbian census results (2011).
URL http://pod2.stat.gov.rs/ObjavljenePublikacije/Popis2011/Knjiga10.pdf

[53] N. Jaleeli, Understanding automatic generation control, IEEE Transactions on Power Systems 7 (3) (1992) 1106–1122. doi:10.1109/59.207324.

[54] National Electrical Manufacturers Association, Volt/VAR Optimization Improves Grid Efficiency.

[55] S.-Y. Yun, C.-M. Chu, S.-C. Kwon, I.-K. Song, J.-H. Choi, The development and empirical evaluation of the korean smart distribution management system, Energies 7 (3) (2014) 1332–1362. doi:10.3390/en7031332.

[56] S. Ha, Y. Kim, L. Le, I. Rhee, L. Xu, A step toward realistic performance evaluation of high-speed TCP variants, Fourth International Workshop on Protocols for Fast Long-Distance Networks (PFLDNet06).

Authors' biographies

Mita Cokic received M.Sc. degree in electrical and computer engineering from Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia. He is currently working toward Ph.D. degree in power, electronics and telecommunication engineering at the University of Novi Sad, Faculty of Technical Sciences and working as teaching assistant at the same faculty.

Ivan Seskar (S'86-M'86-SM'05) is an Associate Director at WINLAB, Rutgers University, New Brunswick, NJ, USA, responsible for experimental systems and prototyping projects. His research interests include distributed systems, future Internet architectures, smart grid, wireless communications, software defined networking and cognitive and software defined radios. He was one of the co-PIs and a lead project engineer for ORBIT testbed, with responsibility for development, integration and deployment of the radio grid emulator system since the testbed was released as a community resource in 2005, and for which the team received. 2008 NSF Alexander Schwarzkopf Prize for Technological Innovation. He is a Member of ACM and a Co-Founder and CTO of Upside Wireless Inc.

Mita Cokic

Ivan Seskar

Highlights

- Presented, SDN based solution meets strict Smart Grid communication requirements.
- The solution is tested on large-scale network with more than 220.000 customers.
- The most important benefit is achieved for two critical Smart Grid applications.