A proposal for explicit communication quality management in IEC 61850

Christof Brandauer, Matthias Herlich
Salzburg Research
{firstname.lastname}@salzburgresearch.at
Salzburg, Austria

Jürgen Resch
Ing. Punzenberger COPA-DATA GmbH
JuergenR@copadata.com
Salzburg, Austria

Abstract

This paper discusses how the performance requirements of IEC 61850-based energy applications can be met in wide-area networking (WAN) scenarios. For many application functions, concrete performance criteria are defined in the IEC 61850 standards. However, these known quantitative quality of service (QoS) requirements are not explicitly modeled in the substation engineering process. Instead, based on the capabilities of the System Configuration description Language (SCL), the current practice is to assign an application's traffic to a specific priority or traffic class. We argue that this indirection makes it difficult to assess whether the performance criteria will presumably be met. In this paper an SCL extension is proposed that enables the direct modeling of communication requirements and thus ties the IEC 61850-based engineering more closely to the engineering of the communication network. This extended model is used to provide QoS assessment to a substation engineer at design time. Finally, it is feasible to automatically derive network measurement/monitoring tasks from the extended SCL so that the fulfillment of the individual communication requirements can be evaluated at runtime. The paper discusses a fully-automated prototype based on a programmable, distributed measurement framework.

1. Introduction

Many modern energy applications have inherent requirements on the communication network they use. If these requirements are not met, an application will not work satisfactorily (if at all). Owing to these requirements, the standard IEC 61850-5 [1] defines performance classes and requirements for message types used in energy applications.

However, in the design of a substation the specific requirements of the traffic flows are not *explicitly* modeled and matched with the resources of the communication network. The substation engineer merely assigns network traffic to one of the eight available Ethernet priorities, the switches handle the traffic according to these priorities, and the engineer assumes ("hopes") that the quantitative requirements of the various flows will indeed be met. Within the traditionally overprovisioned (switched) local-area networks this approach is typically sufficient when combined with basic network engineering guidelines as provided e.g. by IEC 61850-90-4 [2].

However, as IEC 61850-based applications increasingly communicate over wide-area networks (WAN) as well, the situation changes fundamentally. In the WAN, the assumption of highly overprovisioned links is usually not valid. Additionally, the traditional SDH/SONET networks which provide hard Quality of Service (QoS) guarantees "out of the box" are increasingly replaced by packet switched networks – often involving the use of the Internet Protocol (IP) – that require extra measures to attain even soft QoS guarantees.

The recently published TR IEC 61850-90-12 [3] provides "*Wide area network engineering guidelines*". Among other things the comprehensive document describes the nature of the QoS parameters that are essential for many IEC 61850-based applications and enumerates specific performance requirements for functions of the various application groups (e.g. teleprotection, condition monitoring, WAMS, WAMPAC, etc.).

The non-trivial question is how the requirements can be met in a packet-switched wide-area network. In such an operational context, providing QoS guarantees is much more complex than in local Ethernet-based networks. For the WAN, different QoS architectures and technologies (e.g. Differentiated Services, MPLS, MPLS-TE, etc.) are available and they provide many degrees of freedom in their deployment. The common denominator is that packets are not all treated in the same way ("best effort" model). Instead, packets are "marked" and depending on their marking they are then treated differently by the switches and routers along the path. As described in TR IEC 61850-90-12 [3], it is necessary to map these markings as packets travel across technological boundaries, e.g. from a LAN priority (802.1Q [4]) to a Differentiated Services Code Point (DSCP) or from a DSCP to an MPLS Traffic Class field (TC, formerly called EXP field [5]).

There is no universal recipe on what mapping to apply as it depends on what specific technologies a given network operator uses and how they are deployed concretely. The number of available markings may vary along this technology chain, e.g. from 8 (802.1Q [4], TC) to 64 (DSCP). (Although MPLS can also be operated in the so-called L-LSP mode, where the combination of the Label and the TC field is used to enable up to 64 different markings as well.) Even beyond the marking aspect the QoS models are not always a good match. As an example, the strict priority-based scheme of Ethernet has no matching equivalent in the Differentiated Services approach and its currently defined Per-Hop Behaviors.

Assigning traffic to some priority / traffic class merely expresses some *qualitative* notion of importance or criticality of the traffic but there is no guarantee that the *quantitative* performance requirements of the energy application will indeed be met. If, e.g., too much traffic is assigned to a high-priority traffic class, this class gets overloaded and undesired effects are inevitable (potentially high latency / high drop rate / starvation of lower priority traffic / etc.).

To preclude such a situation we studied how the IEC 61850-based engineering of the energy domain can be tied more closely to the engineering of the communication network. The primary goal is to assess at design time whether the quantitative performance requirements of the energy applications are attainable.

2. IEC 61850-based QoS modelling

Nowadays, the engineering processes in the energy and communication domain are usually coupled loosely and informally. Typically, a substation operator requests networking services from a third-party provider or an in-house department. It demands secure connectivity between a number of locations (substations, control center(s)) and defines aggregate performance criteria that have to be met. Based on such a request the network operator engineers its network, establishes the requested services, and assures them contractually in the form of a Service Level Agreement containing a technical Service Level Specification (SLA/SLS). Besides dependability aspects, such an SLS defines (per traffic class) upper limits for performance metrics (e.g. latency, jitter, loss) for a maximum amount of traffic. Excess traffic is typically dropped at the ingress of the core network as it would otherwise degrade the performance of all traffic and QoS levels couldn’t be met anymore.

To prevent such a situation, the substation engineer has to ensure that for each WAN destination the sum of traffic per traffic class remains within the given limits. At the same time, the quantitative performance requirements of the energy application that generates the traffic need to be met. The specific communication requirements of many energy applications are listed in TR IEC 61850-90-12 [3]. They are expressed in terms of network performance parameters. This simplifies the quality provisioning process compared to e.g. multimedia applications, where the requirements are often defined on a perceptual level (“high quality video”) and the process of mapping such demands to network performance metrics is non-trivial.

Nevertheless, the known quantitative QoS requirements are not explicitly described in the substation engineering process, but instead an application’s traffic flow is assigned to a priority level / traffic class. As explained above this indirection makes it very difficult to assess whether the requirements will indeed be met. Also, the current engineering process makes it very hard to predict whether the total amount of traffic remains within the bandwidth limits up to which the network operator guarantees certain levels of performance.

We argue that this indirect management of requirements is a weak spot in the IEC 61850-based engineering process that is particularly relevant in wide-area communication scenarios. To address it, we propose (i) to explicitly include quantitative performance requirements in the modeling process and (ii) to assess whether they can be fulfilled at design time.

2.1 SCL extension

The System Configuration description Language (SCL) as standardized in IEC 61850-6 [6] is used to describe IED configurations and communication systems. In this section it is discussed how the SCL can accommodate the proposed QoS modeling extension.

The SCL has been designed with extensibility in mind and it is thus possible to add new information elements without the need to modify the SCL schema itself. The type `<tBaseElement>`, from which most elements are derived, allows the addition of arbitrary attributes and elements from namespaces other than the default SCL namespace. These additions are, however, generally not preserved when an SCL file is passed between engineering tools of different vendors. This is in contrast to the `<Private>` element, an optional child element of `<tBaseElement>`, which must be preserved at data exchange among tools. This is an important aspect and we therefore choose the `<Private>` element as the extension point and define an XML schema that we use for its content. Our schema defines its own namespace to which we refer to with a namespace prefix starting with the character **e** as required in IEC 61850-6 [6].

Instead of using the `<Private>` element, it would of course also be possible to extend the SCL schema itself to accommodate the proposed QoS modelling extension. This would require an update of the IEC 61850-6 standard and is thus not within the scope of this paper. For the given work, we make use of the existing extension mechanisms and provide our extension as the self-contained file `SCL_QoS.xsd` which is shown in Appendix A.

The schema makes use of existing SCL units. Missing units are added as part of our extension so that it remains self-contained. To this end we added `<tDataSize>` and `<tPercent>` and modelled them similar to existing SCL units.

2.2 Content of the proposed extension

As argued in detail above we propose to add the quantitative performance requirements of a traffic flow to the SCL (e.g. concerning the latency, jitter, or packet loss). Additionally, it is necessary to characterize the traffic for which these performance criteria are requested. To this end, the proposed QoS extension consists of a root `<QoS>` element which contains 2 child elements called `<TrafficSpec>` and `<QoSRequirements>`.

For the characterization of the traffic we make use of the well-known TSpec of Integrated Services as defined in RFC2215 [7]. The TSpec is based on an enhanced token bucket model. A flow is characterized by the following parameters:

- average data rate,
- bucket depth,
- peak data rate,
- maximum packet size, and
- minimum policed unit.

The original TSpec uses bytes as the unit for data sizes but the SCL uses bits to describe data sizes and we thus stick to the SCL units.

The `<QoSRequirements>` element used for the specification of performance requirements contains the relevant performance metrics, e.g. latency, jitter, and packet loss. Additional metrics can be added as needed.

2.3 Selection of the proper parent element

The substation engineer shall be able to specify the performance characteristics at the granularity of a single traffic flow. Therefore, the extension is added to the control block of the communication section. The control block of a `<ConnectedAP>` contains the `<Address>` which contains performance related configuration like the VLAN-PRIORITY. However, as this element is also used to specify the configuration of an interface for which the WAN performance specifications are meaningless, it is not a proper parent element.

To cleanly separate performance from addressing, we add our performance extension to the `<GSE>` and `<SMV>` elements, respectively. Adding the extension there allows the specification of performance requirements for each flow and still keep them separated from addressing aspects. The extension we propose in this paper is most useful, when the flows that the `<GSE>` and `<SMV>` elements describe are transmitted over a WAN network as described in IEC 61850-90-5 [8].

2.4 Example

To illustrate the use of the proposed QoS extension, we applied it to the example file shown in Annex B of IEC 61850-90-5 [8]. Listing 1 shows an excerpt.

```
<ConnectedAP iedName="AA1FP1" apName="S1">
  <Address>
    <P type = "IP">172.16.1.3</P>
    ...
  </Address>
  <SMV desc="Phasor SVCB" ldInst="PMU" cbName="SyPh_SVCB1">
    <Private type="SalzburgResearch:QoS">
      <extQoS:QoS xmlns:extQoS="http://salzburgresearch.at/ns/61850/QoS/v1">
        <extQoS:TrafficSpec>
          <extQoS:AverageDataRate unit="b/s" multiplier="M">0.1</extQoS:AverageDataRate>
          <extQoS:BucketDepth unit="b" multiplier="k">2112</extQoS:BucketDepth>
          <extQoS:PeakDataRate unit="b/s" multiplier="M">0.1</extQoS:PeakDataRate>
          <extQoS:MaximumPacketSize unit="b">1056</extQoS:MaximumPacketSize>
          <extQoS:MinimumPolicedUnit unit="b">1056</extQoS:MinimumPolicedUnit>
        </extQoS:TrafficSpec>
        <extQoS:QoSRequirements>
          <extQoS:Latency unit="s" multiplier="m">10</extQoS:Latency>
          <extQoS:Jitter unit="s" multiplier="m">0.1</extQoS:Jitter>
          <extQoS:PacketLoss unit="%">0</extQoS:PacketLoss>
        </extQoS:QoSRequirements>
      </extQoS:QoS>
    </Private>
    <Address>
      <P type = "IP">172.16.0.100</P>
      ...
    </Address>
  </SMV>
</ConnectedAP>
```

Listing 1 Excerpt of an SCL file making use of the proposed QoS extension

3. Exploitation of the extended SCL model

3.1 Performance assessment

Enumerating the traffic flows with their bandwidth and QoS requirements has no use per se. The primary purpose is to evaluate whether these requirements can be fulfilled with the resources provided by the communication network. Ideally, the substation engineer can be supported in answering that question during the design phase within his/her tool environment. It shall be possible to make this assessment in the absence of online interaction with the network provider but simply based on a static specification of the network services.

As described in the beginning of this section, a network provider establishes communication services upon an initial request by a customer. How this is achieved will vary greatly between providers and the details of a concrete implementation (LSPs, DiffServ Per-Hop Behaviours, resource reservations, internal topology and routing, etc.) are normally not disclosed. From the point of view of the customer, the core network looks like an opaque cloud that provides connectivity (for multiple traffic classes) between the distributed locations. This situation is illustrated in Figure 1 (although in the drawing the cloud is not completely opaque to indicate to the reader that the network does contain internal routers).

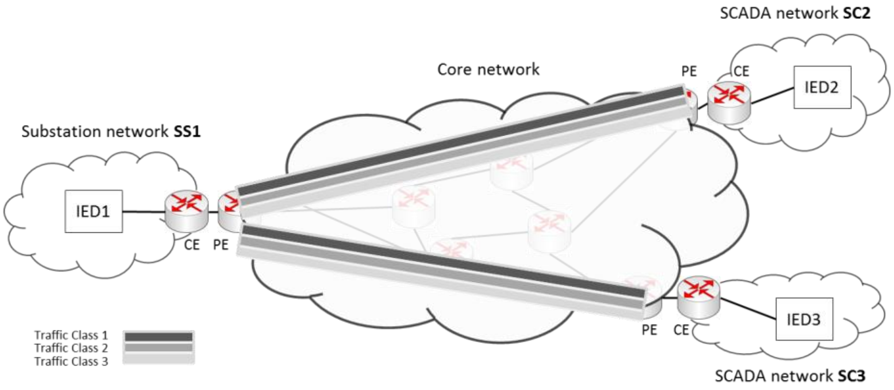

Figure 1 WAN connectivity from the customer's point of view

The technical details of the services that are contractually agreed between the network provider and the customer are specified in a SLS. Conceptually, such an SLS contains information as shown in Table 1. For easier reading, the source and destination address prefixes have been replaced by short names.

	Traffic Class	Source	Destination	Bandwidth	Latency	Jitter	Packet loss
■	TC1	SS1	SC2	4 Mbit/s	10 ms	2 ms	0 %
▒	TC2	SS1	SC2	10 Mbit/s	50 ms	5 ms	0 %
░	TC3	SS1	SC2	50 Mbit/s	500 ms	20 ms	0.1 %
■	TC1	SS1	SC3	2 Mbit/s	10 ms	2 ms	0 %
▒	TC2	SS1	SC3	20 Mbit/s	50 ms	5 ms	0 %
░	TC3	SS1	SC3	30 Mbit/s	500 ms	20 ms	0.1 %

Table 1 Exemplary Service Level Specification (SLS)

The provider's responsibility is to ensure that these services are continuously delivered. On the other hand, the substation engineer has to make sure that the quantitative performance requirements of the energy applications can be fulfilled on top of these communication services.

To address this performance assessment task, we propose a component called a *Resource Mapper*. As shown in Figure 2, its inputs are (i) the `<QoS>` elements as provided within the extended SCL and (ii) the specification of the network services (SLS). The Resource Mapper's job is to find a feasible mapping from the individual traffic flows to the available communication services.

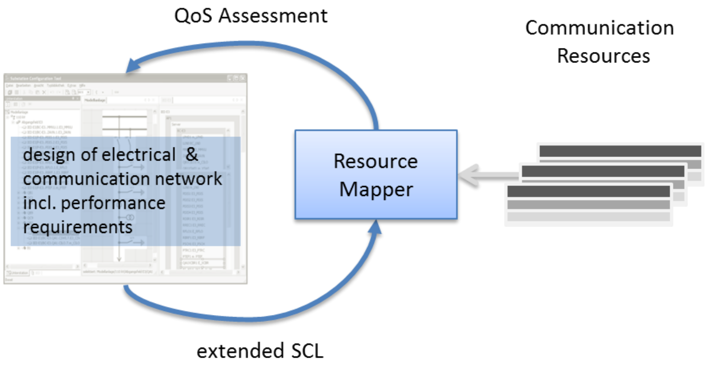

Figure 2 Performance Assessment by the Resource Mapper

Based on the destination address prefix and the performance requirements of an individual flow, the Resource Mapper searches the SLS table for a traffic class that can fulfill these requirements. If such a mapping can be found for each WAN flow of the extended SCL, the second step is to evaluate for each available traffic class whether the flow aggregate that is assigned to this class fits within its bandwidth limit. For a simple worst case calculation, the peak rates of the flows are summed. Depending on the permitted loss ratio and the ratio between peak and average data rates, more complex bandwidth models might be applicable and the Resource Mapper can be extended accordingly.

3.2 Traffic marking configuration of egress traffic

Before a substation's WAN traffic leaves the substation network via the Customer Edge (CE) router, each packet has to be assigned to the proper traffic class. The TR IEC 61850-90-12 states that this is accomplished by an application-aware instance like the Proxy-Gateway or the CE.

Note that with our proposed approach the Resource Mapper computes a mapping from the applications' quantitative requirements to the traffic classes provided by the communication network. The primary goal of our proposed extension is to assess whether all requirements can be supported. However, as a side product, the Resource Mapper automatically produces exactly the type of mapping specification that is required for the configuration of the marking mechanisms in the Proxy-Gateway/CE.

3.3 Automated performance monitoring

The extended SCL contains the specification of all WAN traffic flows and their performance requirements. The Resource Mapper provides the rules for mapping those flows to the Traffic Classes supported by the communication network. The availability of this information enables the implementation of an automated quality monitoring/measurement system. It can be used to automatically derive the monitoring tasks that are required to assess at runtime, whether the performance criteria of the individual flows are indeed met.

We propose a 2-stage process: in the first stage, abstract monitoring tasks are derived. They specify what metrics have to be measured on which network segments and what the performance thresholds are. These tasks are abstract in the sense that they specify *what* has to be measured but not *how* it has to be done. Also, these tasks are not tied to any specific monitoring platform.

In the second stage, these abstract tasks are transformed into concrete tasks that can be executed on a specific monitoring platform. This transformation step is specific to the target platform that executes the measurements and considers the specific constraints of a given installation, e.g. the location of measurement agents and the types of permissible measurement strategies (e.g. active / passive).

WAN measurements often require the deployment of multiple distributed measurement points that need to be controlled and orchestrated (distribution of measurement task; initialization; execution of the measurement task; error handling; etc). The measurement of a one-way latency usually requires precise time-synchronization between the measurement points. Any capable measurement platform can be used to carry out the measurements, e.g. an Omicron DANEO 400[1]. In the next section we describe a prototype implementation that makes use of our own measurement platform MINER.

4. Performance monitoring prototype

As a proof of concept, we realized a prototype that implements a fully automated performance monitoring system.

The prototype is based on the "*Modular Infrastructure for Networked Experimentation and Research*" (MINER) [9]. It was developed at Salzburg Research over the course of several European research projects (e.g. AQUILA, INTERMON, MOME, NETQOS) in the thematic area of QoS for IP networks and has since then been used in various research activities and industry projects. The main objective of MINER is to support users in carrying out distributed experiments in communication networks. To this end, MINER enables a user to define complex experiments where multiple "tools" have to be executed on remote hosts in a coordinated fashion. A "tool" performs some kind of activity, e.g. the generation of a traffic flow, an active or passive performance measurement, etc. In many cases, the

---

[1] www.omicron.at/daneo400

activities are performed by existing binaries which can easily be converted into MINER tools. Some other tools were developed from scratch, notably those that perform high-precision IP performance measurements of one-way delay/jitter/loss/duplication/reordering according to the IPPM standards [10] [11] [12] [13] [14].

A MINER installation consists of a central *Server* and so called *ToolProxies*. On each node where an activity is going to be executed as part of the experiment, a ToolProxy is deployed. It provides the execution environment for the *MINER Tools* and proxies the communication between them and the Server. Figure 3 shows an exemplary deployment.

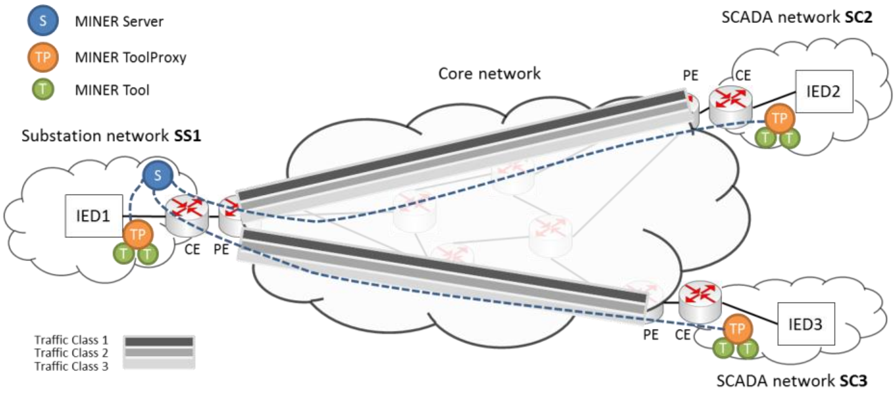

Figure 3 Exemplary MINER deployment

MINER is designed as a modular system and it can be extended with arbitrary tools via its `ToolAPI`. For the user of the platform, the complete functionality (specification / scheduling / controlling of experiments, retrieval of results, etc.) is available via a webservice-based `UserAPI`. The programmability and extensibility of the system make it a good candidate for the realization of the automated performance monitoring prototype.

The prototype realizes the following steps:

(1) transformation of the extended SCL to abstract measurement tasks

This step is straight forward. The extended SCL contains the specification of the traffic flows and their performance requirements in a well-defined XML-based format. The relevant information is extracted and converted to a simple XML file describing the abstract measurement tasks.

(2) transformation of abstract measurement tasks to a MINER experiment

This step is naturally specific to the MINER platform. Besides requiring knowledge about the MINER platform in general (how to specify an experiment etc.), it is necessary to have knowledge about the concrete installation, e.g. where the MINER ToolProxies are located and what type of measurements they can perform. The output of this processing step is the specification of a MINER experiment (coincidentally an XML file again).

(3) execution of the MINER experiment

The experiment specification is submitted to the MINER Server via the MINER `UserAPI` and it can then be scheduled or immediately executed.

(4) evaluation of the performance criteria

MINER enables the definition of conditions on the measured metrics (e.g. delay < 50ms). In that case, a ToolProxy detects the violation of a condition and emits an alarm (which can be handled through a plugin at the Server). If this facility is not used, the measured results can simply be retrieved via the `UserAPI` and compared with the requested performance criteria.

The prototype has been validated in the context of a synchrophasor WAN transmission according to TR IEC 61850-90-5. The test setup consisted of a GE N60 PMU[2] located in Salzburg that transmitted the synchrophasors to a SCADA system (Zenon energy edition) installed in Berlin. The WAN connection contained 2 microwave links in a private network and a segment through the public

---

[2] https://www.gegridsolutions.com/multilin/catalog/n60.htm

Internet. The setup was purely used for test purposes and there was no real application behind it. For our assumed “under-voltage load shedding” use case we defined performance requirements w.r.t. latency, jitter, and loss. Packet markings were not considered as only a single traffic class was available. The prototype was tested with 2 different configurations: in one case, the abstract monitoring tasks were mapped to passive measurements based on sniffing network traffic; in the other case, they were mapped to light-weight active measurements according to the IPPM standards.

5. Conclusions and future work

It this paper it was analyzed how the performance criteria of IEC 61850-based energy applications can be met when they are employed over wide-area networks. In such a scenario the task of guaranteeing certain QoS levels is significantly more complex than in the typically overprovisioned local area networks. We proposed an SCL extension that enables a substation engineer to work directly with the quantitative performance requirements of the applications. This engineering tool independent approach avoids the loss of valuable information that occurs in the current practice where requirements are mapped to priorities / traffic classes.

Based on the extended SCL and a service level specification of the communication network operator, we designed a prototype that provides QoS assessment to the engineer at design time. As an additional benefit, it is shown how the proposed extension can be exploited to implement of fully-automated monitoring system that evaluates at runtime whether the application’s performance criteria are indeed met.

The next step towards practical use of the proposed extension is to extend a System Configuration Tool to generate an SCL that includes performance requirements (so far it was done manually in the SCL file). To test how flexible our approach for performance monitoring is, it could be implemented using a different monitoring platform. Also, moving from a pure test environment to a real-world deployment will yield more insight into the practical applicability.

Acknowledgement

This work is funded by the Austrian Ministry for Transport, Innovation and Technology (bmvit) and the Climate & Energy Fund under the "Energy Mission Austria" program in the IPEN project (FFG No. 838626).

References

[1] *IEC 61850-5: Communication networks and systems for power utility automation - Part 5: Communication requirements for functions and device models,* 2013.

[2] *IEC TR 61850-90-4: Communication networks and systems for power utility automation - Part 90-4: Network engineering guidelines,* 2013.

[3] *IEC TR 61850-90-12: Communication networks and systems for power utility automation - Part 90-12: Wide area network engineering guidelines,* 2015.

[4] *IEEE Std 802.1Q: Bridges and Bridged Networks,* 2014.

[5] L. Andersson and R. Asati, *RFC 5462: Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field,* 2009.

[6] *IEC 61850-6 International Standard: Communication networks and systems for power utility automation - Part 6: Configuration description language for communication in electrical substations related to IEDs,* 2009.

[7] S. Shenker and J. Wroclawski, "RFC 2215. General characterization parameters for integrated service network elements," 1997.

[8] *IEC 61850-90-5 Technical Report: Communication networks and systems for power utility*

automation - Part 90-5: Use of IEC 61850 to transmit synchrophasor information according to IEEE C37.118, 2012.

[9] C. Brandauer and T. Fichtel, "MINER - A Measurement Infrastructure for Network Research," in *Proceedings of the 5th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, 2009.

[10] G. Almes, S. Kalidindi and M. Zekauskas, *RFC 2679: A One-way Delay Metric for IPPM,* 1999.

[11] C. Demichelis and P. Chimento, *RFC 3393: IP Packet Delay Variation Metric for IP Performance Metrics (IPPM),* 2002.

[12] G. Almes, S. Kalidindi and M. Zekauskas, *RFC 2680: A One-way Packet Loss Metric for IPPM,* 1999.

[13] H. Uijterwaal, *RFC 5560: A One-Way Packet Duplication Metric,* 2009.

[14] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov and J. Perser, *RFC 4737: Packet Reordering Metrics,* 2006.

Appendix A: XML Schema of the proposed QoS extension

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="http://salzburgresearch.at/ns/61850/QoS/v1"
  xmlns="http://salzburgresearch.at/ns/61850/QoS/v1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:scl="http://www.iec.ch/61850/2003/SCL"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">Schema for a Quality of Service (QoS)
extension.</xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://www.iec.ch/61850/2003/SCL" schemaLocation="SCL_BaseTypes.xsd"/>

  <xs:element name="QoS">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="TrafficSpec" type="tTrafficSpec" />
        <xs:element name="QoSRequirements" type="tQoSRequirements" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="tTrafficSpec">
    <xs:sequence>
      <xs:element name="AverageDataRate" type="scl:tBitRateInMbPerSec" />
      <xs:element name="BucketDepth" type="tDataSize" />
      <xs:element name="PeakDataRate" type="scl:tBitRateInMbPerSec" />
      <xs:element name="MaximumPacketSize" type="tDataSize" />
      <xs:element name="MinimumPolicedUnit" type="tDataSize" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="tQoSRequirements">
    <xs:sequence>
      <xs:element name="Latency" type="scl:tDurationInMilliSec" minOccurs="0" />
      <xs:element name="Jitter" type="scl:tDurationInMilliSec" minOccurs="0" />
      <xs:element name="PacketLoss" type="tPercent" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="tDataSize">
    <xs:simpleContent>
      <xs:extension base="xs:nonNegativeInteger">
        <xs:attribute name="unit" type="xs:normalizedString" use="required" fixed="b" />
        <xs:attribute name="multiplier" type="scl:tUnitMultiplierEnum" use="optional" default=""/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="tPercent">
    <xs:simpleContent>
      <xs:extension base="tZeroToHundred">
        <xs:attribute name="unit" type="xs:normalizedString" use="required" fixed="%" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:simpleType name="tZeroToHundred">
    <xs:restriction base="xs:decimal">
      <xs:minInclusive value="0" />
      <xs:maxInclusive value="100" />
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```