# On the Necessity of Traffic Shaping for PMU Measurement Data Streams *

Miroslav Popovic, Peng Gao, Dan-Cristian Tomozei, Jean-Yves Le Boudec
EPFL, IC-LCA2, Switzerland
{miroslav.popovic, p.gao, dan-cristian.tomozei, jean-yves.leboudec}@epfl.ch

## ABSTRACT

We study the communication network of an on-campus active distribution network (ADN) that comprises phasor measurement units (PMUs) connected to medium-voltage transformers. Within stringent time delays and with minimal loss the PMUs periodically transfer fresh measurements through a phasor data concentrator (PDC) to a centralized electrical network state estimator (control point). Due to strict operational constraints, a dedicated robust communication infrastructure that withstands power shortages is needed. We use DSL technology (SHDSL) over existing telephone cables.

We investigate the operating region of the system. In our experimental setup, we measure the achieved goodput (application layer throughput) for various measurement frequencies and frame sizes. We observe that goodput drops catastrophically, in some scenarios for which the PDC data transmission rate slightly exceeds the capacity of the SHDSL link for a short period of time. Specifically, when the offered traffic exceeds channel capacity by 20%, we observe up to 90% of lost packets.

We explain this surprising phenomenon by the combination of IP fragmentation that splits each frame in two IP packets of significantly different sizes and the FIFO/tail-drop queuing discipline implemented within line terminal devices at the source end. We conclude that the guidelines of the C37.118.2-2011 standard are not sufficient for designing a PMU data transfer layer. Implementing traffic shaping within PMUs is essential for avoiding excessive packet losses.

## 1. INTRODUCTION

Active distribution networks (ADNs) are characterized by high penetration of renewable energy generation (e.g., PV roofs). Due to the volatility of renewable power injection (e.g., 1/20 mean-to-peak ratio for solar cells in Switzerland), such networks are highly dynamic. The real-time operation of an ADN uses its knowledge of the network global state, which requires performing quasi-simultaneous, accurate, high-frequency measurements of physical quantities (such as voltage and phase) in buses of the electrical network. They are obtained by Phasor measurement units (PMUs) equipped with synchronized clocks (e.g., via GPS) that are connected to medium-voltage transformers. The data are then conveyed via a communication network to a central control point for processing by an electrical network state estimator. Based on the estimated electrical network state, the network controller takes control actions that aim to keep the electrical network within a predefined (safe) range of operating set-points.

As the controller relies on up-to-date state estimates, the communication network plays a crucial role in the operation of an active distribution network. It needs to ensure the delivery of PMU data within stringent time delays and with minimal loss to the central control point. Additionally, it needs to be prone to power cuts, because at such critical moments its availability is essential. For these reasons alone, a dedicated communication infrastructure is required (as typical communication networks are inoperative during blackouts).

It is common practice to aggregate PMU data (e.g., from PMUs that are found in nearby locations) by using a so-called phasor data concentrator (PDC). This equipment collects PMU data and transmits it downstream.

Figure 1 gives us an idea of the communication network that will be deployed as a part of the EPFL campus-wide smart-grid infrastructure. We use DSL technology (SHDSL) over passive twisted pair cables.

The goal of the work presented in this paper is to define the operating region of the wired infrastructure. To understand the physical layer limits of the SHDSL communication channels that are to be deployed we investigate the behavior of the network in situations of congestion when the load is slightly above the capacity limit for short periods of time, as non-determinism that characterizes software used in network components can cause unexpected behavior. For example, packets containing measurement results might be backlogged due to a software update or a scheduled security session key replacement. As a result, there occurs a burst of packets that, for a short period of time, requires a larger ca-

*This research has received funding from the EU 7th Framework Programme (FP7/2007-2013) under grant agreement n. 257740 (Network of Excellence ”TREND”).

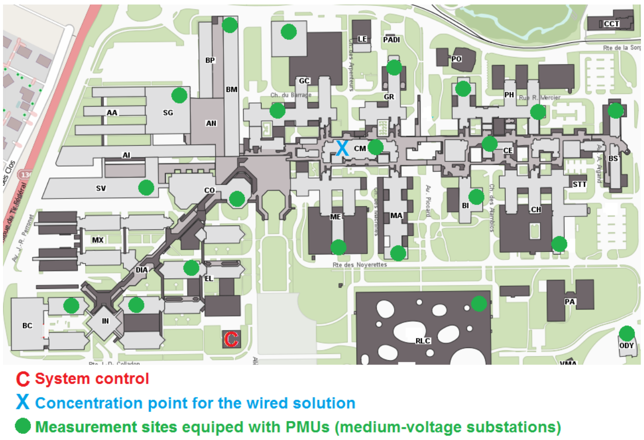



Figure 1: EPFL campus map with the important points, connected with wired infrastructure. We use the existing telephone infrastructure (twisted-pair cabling). This system is passive and interconnects all the campus buildings with the control point.

pacity than available (which implicitly results in packet losses). Nevertheless, even when packet losses are inevitable, we need to ensure that they are not excessive. In this paper we show that with off-the-shelf equipment excessive packet losses are possible in some practical scenarios. Hence, the design phase has to encompass this issue.

Our approach is to begin by measuring the maximum throughput that can be achieved on an SHDSL communication channel. When we conducted further tests they led to surprising results. We observe that when the PDC data transmission rate slightly exceeds the capacity of the SHDSL link, even for a short period of time, goodput drops catastrophically. Specifically, we measure a maximum achievable goodput of 1.98Mbps at the destination (the capacity of the SHDSL link). However, when the PDC generates data at 2.2Mbps, we obtain drastically different goodput at the destination, depending on the frame size: for a frame size of 2178, the goodput is 1.25Mbps, i.e., 37% loss, whereas for a frame size of 1815B, the goodput is 850kbps, i.e., 57% loss. These findings are supported by simulations.

We explain this phenomenon by the combination of two separate factors: IP fragmentation that splits each frame in two IP packets with very different sizes, and FIFO/tail-drop queuing discipline within line terminal devices at the source end. We conclude that following the guidelines from C37.118.2-2011 standard [1] is not sufficient for designing a PMU data transfer layer. It comes from the fact that the standard defines messaging including types, use, contents and data-formats, whereas traffic management issues are out of its scope. In the concluding section we give some guidelines on how the problem can be mitigated by implementing traffic shaping within PMUs.

The paper is organized as follows. Test setup and measurement results are presented in Section 2. Our simulation setup and simulation results are described in Section 3. Concluding remarks with design guidelines are given in Section 4.

# 2. EXPERIMENTS

## 2.1 Experimental setup

The experimental setup is depicted in Figure 2.

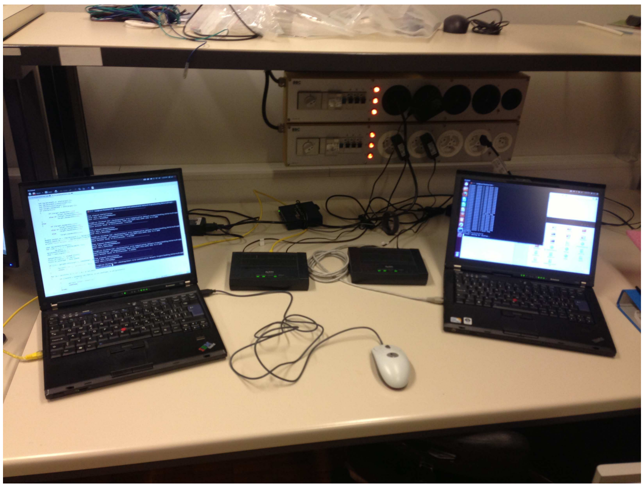


Figure 2: Experimental setup. SHDSL line terminals (black boxes in the middle) are connected by a twisted-pair loopback circuit.

We use ZyXEL SHDSL line terminals [2] that implement G.SHDSL.bis technology [3]. They are connected via a twisted-pair loopback circuit. The length of the twisted-pair corresponds to the typical distance between two end points on campus. The two PCs run Ubuntu Linux. They are connected to SHDSL line terminals and are used for several purposes: to emulate PDCs/PMUs, to run tools like `iperf` and `ping`, or to run other custom applications written in C++ designed for a specific experiment. The connection between each PC and its line terminal is a 100Mbps Ethernet link. The line terminal forwards the data on the twisted-pair.

The very first experiments evaluate maximum available throughput that can be achieved, as well as the round-trip time (RTT). To this end we used the standard `iperf` and `ping` tools, respectively. On average we found a maximum available throughput of 1.98 Mbps and a RTT of 3.4 ms.

## 2.2 Discovering line terminal queue size and queuing discipline

Due to the lack of documentation from the manufacturer of SHDSL line terminals, we perform experiments to discover the characteristics of the queues that are implemented within a device. This was necessary in order to understand the packet-loss patterns presented in the next subsection. The input interface capacity is roughly 50 times higher than the output interface capacity. The bottleneck where packets are dropped is a queue that corresponds to the outgoing interface of the line terminal (see Figure 3).

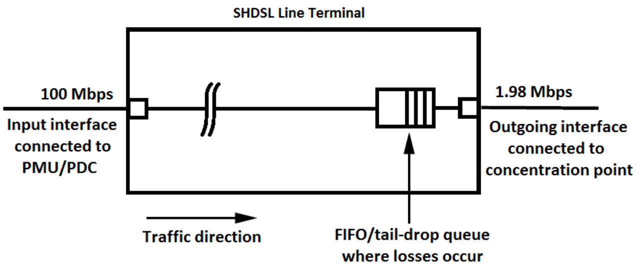



Figure 3: FIFO/tail-drop queue within SHDSL line terminal where losses occur.

We run a sequence of experiments to determine the queue size and the queuing discipline. On the sender side we send bursts of packets, and we examine the received packets on the receiver side. We vary the burst size between 1 and 100 packets. We also vary the size of the payload (50 B, 500 B, 1000 B and 1452 B). Each packet that is sent contains a sequence number that is inspected at the receiving end. Results of these experiments are depicted in Figure 4.

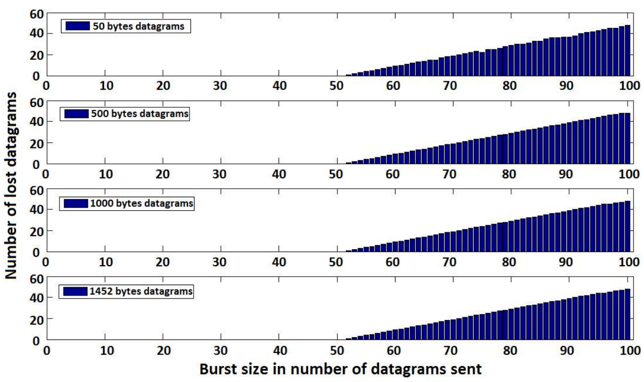



Figure 4: Number of lost packets for different datagram and burst sizes.

We observe that regardless of the packet size, the 52$^{\text{nd}}$ packet is always the first one to be lost. Furthermore, by analyzing the sequence numbers of the received packets, we observe that all the following packets are also lost. We conclude that the outgoing buffers associated with SHDSL line-terminal interfaces are implemented as FIFO queues with tail-drop queue management algorithm, with a queue size of 50 packets.

### 2.3 When fragmentation occurs, information loss might be significantly above expected.

As mentioned in the introduction, it is important to investigate the behavior of the network in situations when the load, for short periods of time, is slightly above the link capacity limit. Although in this case packet losses cannot be avoided, we aim to quantify how many packets are discarded.

The role of a PDC is to aggregate measurements - received from a number of PMUs or other PDCs - to a single stream that is then forwarded to the control point. Measurements are correlated by time-tags. Depending on the number of streams that are aggregated, the resulting UDP datagrams might exceed the threshold size beyond which IP fragmentation is required.

This size is dictated by the maximum transmission unit (MTU) of the underlying protocol. In our case, we use Ethernet that has a MTU of 1500 B. If the total size of the UDP payload, the UDP header (8 B) and the IPv6 header (40 B) is above 1500 B, IP fragmentation will take place, and one UDP datagram will be encapsulated in two (or more, if necessary) IP packets. At the receiver's side a UDP datagram can be reassembled only if all IP packets that carry its fragments are correctly received.

If any fragment is lost, the whole UDP datagram is considered as lost since the transport layer is not able to reassemble the datagram. In a toy example in Figure 5, we compare consequences of two different loss patterns that are due to two different scheduling strategies in the bottleneck queues. These are the best-case and the worst-case loss patterns. In both cases the same fraction of bits is discarded. In the best case the overall information loss equals the fraction of lost bits, whereas in the worst case all the information is lost.

This suggests that when UDP datagrams are fragmented in two IP packets, an optimal scheduler should not discard only one of the two corresponding IP packets. The transmission of the other would use network resources without any benefit, as it would be discarded anyway at the receiver side. Hence, if one of the UDP datagram fragments is discarded, the optimal scheduler would also discard all other IP packets that carry fragments of the same UDP datagram. A lower bound on the resulting loss probability in the case when the offered traffic is above line capacity is

$$\text{loss probability} \geq \frac{\text{offered traffic} - \text{line capacity}}{\text{offered traffic}} \quad (1)$$

In other words, ideally if the offered traffic is only slightly above the channel capacity, only a small fraction of information should be lost.

We perform experiments to measure how close the system is to this optimal loss probability, given the equipment and technology at our disposal. The IEEE Standard C37.118.2-2011 states that a variable num-

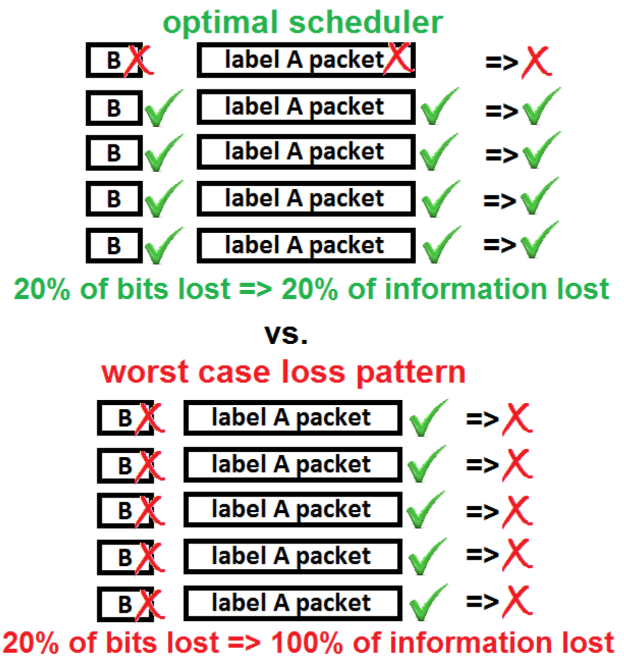
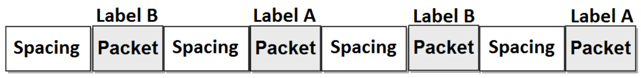



Figure 5: Toy example. Label A and Label B packets represent two fragments of the the same UDP datagram. Depending on the loss pattern the effect at the receiver might be dramatically different.

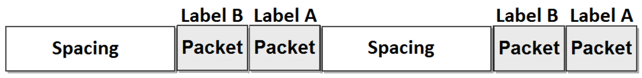
ber of PMU measurements can be included within a single frame. We develop a PDC traffic emulator that allows us to vary the size of the packets on the sender side. We examine two sending patterns. *Sending pattern I* (Figure 6(a)) mimics the scenario where no IP fragmentation occurs. One UDP datagram results in one IP packet and there is always time spacing between two consecutive packets. Beginning with the target throughput, we calculate this time spacing duration between two consecutive packets. *Sending pattern II* (Figure 6(b)) emulates the situation when IP fragmentation occurs. In this case, one UDP datagram results in two IP packets that are sent back-to-back. As in the first case, waiting times (now between groups of two IP packets) are calculated to meet the target throughput. In both cases alternate packets are labeled with $A$ or $B$ and Label $A$ packets are sent first, Label $B$ packets follow (see Figure 6).

At the destination end, we are interested in the number of successfully received bytes as seen by transport layer for various target throughputs and packet sizes. We keep track of the number and labels of lost packets.

We use the UDP transport layer protocol because TCP retransmits lost packets, which is in our case superfluous as fresh measurements are generated at a high frequency. For every experiment, we send 10000 IP packets labeled with A and 10000 packets labeled with B and we specify sending pattern (I or II), size of pack-




(a) Pattern I (with spacing), Label A packet first then Label B packet.




(b) Pattern II (back to back), Label A packet first then Label B packet.

Figure 6: Sending patterns.

ets labeled with A/B, and the target throughput.

We consider three datagram sizes (and thus implicitly three different label $B$ packet sizes). Measurement results are depicted in Figures 7, 8, and 9. Each point is obtained via a single experiment. For each considered datagram size, we plot on the left panel the number of successfully transmitted Label $A$ and $B$ packets and on the right panel, the number of successfully decoded bytes at the transport layer (UDP datagram payload). We also plot the curve that corresponds to the theoretical minimal-loss probability (labeled “optimal”) in terms of number of bytes, as expressed in Equation 1. In the case of sending pattern I, measurement results for UDP datagram payload follow very well the optimal curve. However, for sending pattern II, we observe that the UDP datagram payload drops significantly. We notice that the number of Label B packets lost is also dramatically higher than for pattern I.

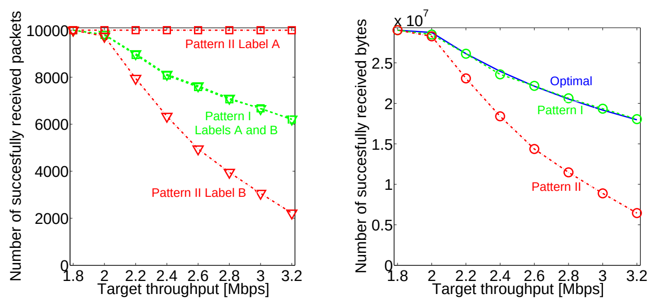



Figure 7: Experimental results. Label A packet size: 1452 B, Label B packet size: 1452 B. 10000 packets of each kind are sent.On the left: number of successfully received packets. On the right: number of successfully received bytes as seen by transport layer.

## 2.4 Smaller Label B packets are more likely to be discarded

Another observation we can make from Figures 7, 8, and 9 is that when sending pattern II occurs, the smaller Label B packets are, the more of them we lose. For example, when the target throughput is 2.4 Mbps ($\sim$

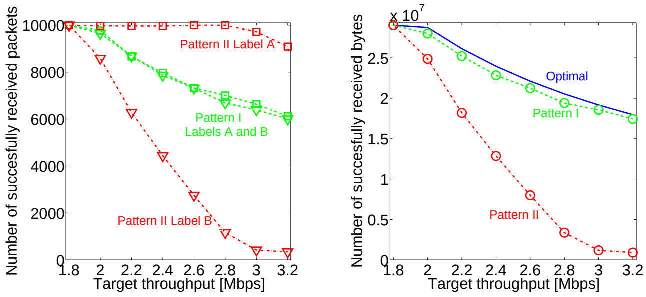



Figure 8: Same as Figure 7, except for Label B packet size: 726 B.

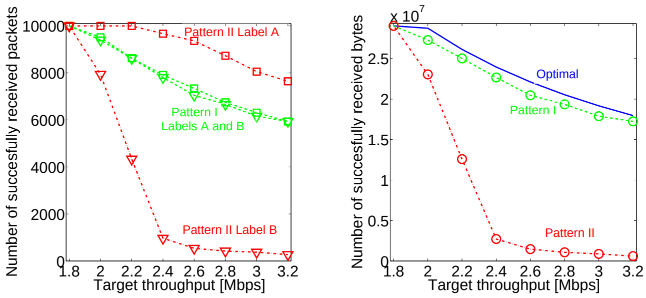



Figure 9: Same as Figures 7 and 8, except for Label B packet size: 363 B.

20% over the SHDSL link capacity), and we set Label B packet size to 1452B, 726B, and 363B, we lose 37%, 56%, and 93% of Label B packets, respectively. This is contrary to what might be expected as smaller packets require less time to be processed.

### 2.5 High Label B loss probability leads to high loss of information

Keeping in mind the discussion about IP fragmentation from subsection 2.2 and by examining the results presented in Figures 7, 8, and 9 when sending pattern II occurs, we remark the correlation between Label B packet loss and the significant drop in goodput. As a result, the number of UDP datagrams that can be used by the receiver drops dramatically. For example, when the target throughput is 2.4 Mbps ($\sim$ 20% over the SHDSL link capacity) and when Label B packet size is 363B, we can only retrieve up to 7% of the measurement results that are sent.

In the following section, for sending pattern II, we compare experimental results with simulation results, and we explain the phenomenon.

## 3. SIMULATIONS

In order to verify and explain the result detailed in Section 2, we design a simulation program. We use discrete event simulation to describe arrivals and departures from a queue that corresponds to the outgoing interface of the line terminal. Based on the experimental results from Section 2.2 we set the queue size to 50 and the queuing discipline to FIFO/tail-drop.

### 3.1 Simulation results match experimental results

Throughout this section, we analyze the results for sending pattern II as we identified in previous section this sending pattern as the critical one.

As mentioned before the first goal of having a simulator is to verify what we observed in our experiments. To this end we show in Figures 10, 11 and 12, a comparison of the results already shown in Figures 7, 8 and 9 with the results we obtain from our simulator for sending pattern II (back to back). We observe that, in all scenarios, measurement and simulation results are in accordance with each other.

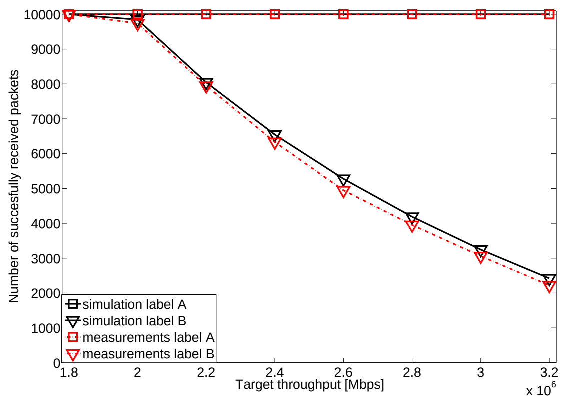



Figure 10: Simulation vs measurement results for sending pattern II (back to back). Label A packet size: 1452 B, Label B packet size: 1452 B. 10000 packets of each kind are sent.

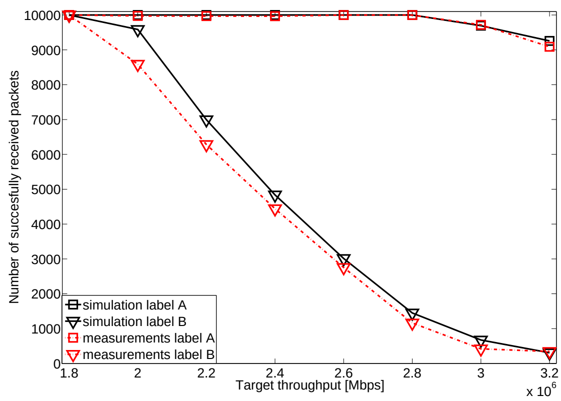



Figure 11: Same as Figure 10, except for Label B packet size: 726 B.

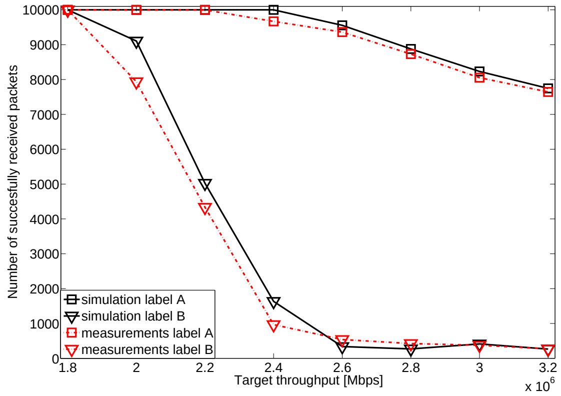
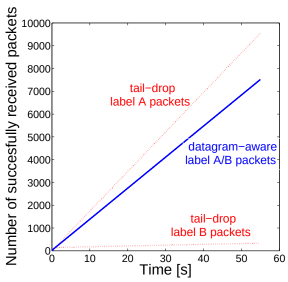


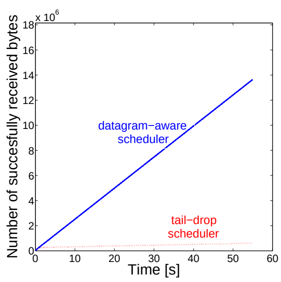

Figure 12: Same as Figures 10 and 11, except for Label B packet size: 363 B.

## 3.2 Small buffers with tail-drop policies cause undesired loss patterns

We use our discrete-time FIFO/tail-drop simulator to better understand the nature of the phenomenon. We show in Figure 13(a), for a specific scenario (details in the figure caption), the evolution over time of the cumulative number of packets of each type that are successfully received when transmission follows pattern II. The plotted trend is similar in all scenarios with sending pattern II. We observe that, after an initial transient period, almost no Label B packets are successfully transmitted. Figure 13(a) also shows that, even when starting with an empty queue, it takes less then 1 second for Label B packets to start being excessively dropped. This shows that Label B packets are systematically discriminated against and that, after a short transient period, very few go through. Whereas, Label A packets receive much better treatment.




(a) Evolution of number of packets of each type that are successfully transmitted.




(b) Evolution of number of successfully received bytes as seen by transport layer.

Figure 13: Simulation results. Comparison of two scheduling policies. Datagram A size is 1452B, Datagram B size is 363, Target throughput is 2.6 Mbps, sending pattern is II.

In Figure 14 we show a possible occupancy of the queue at the arrival moment of two back-to-back packets. There is room for only one packet. The tail-drop queuing policy accepts the first packet and imposes that the second be dropped. We conjecture that this is a typical situation that occurs (i.e., one free slot in the queue) and this situation results in Label B packets always being dropped as they follow Label A packets.

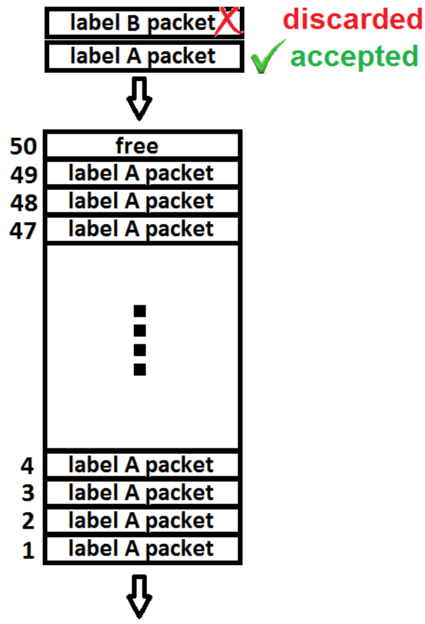



Figure 14: Conjectured typical situation at the FIFO/tail-drop queue that corresponds to the outgoing interface of the line terminal when the load is slightly above the available throughput and once the transient period is finished. Label A and Label B packets that arrive consecutively see a queue that can accommodate only the first one (Label A). This will result in almost no Label B packets in the queue, i.e., almost no Label B packets successfully transmitted.

We use the same parameters to simulate a system, with a difference that we replace the tail-drop queuing policy with the datagram-aware scheduler: If a Label B packet cannot be accepted, the scheduler also discards the corresponding Label A packet, because they both carry fragments of the same UDP datagram. The expected effect is similar to the expected behaviour of an optimal scheduler as described in Section 2. We observe that the resulting effect corresponds to the theoretical optimum as expressed in Equation 1.

In Figure 13(b) we see the number of successfully received bytes as seen by transport layer. If we compare two scheduling policies, we observe that the undesired symptom disappears when we replace the tail-drop queuing policy with our datagram-aware scheduler. In the case of the tail-drop queuing policy the receiver would be able to retrieve around to 3% of the datagrams, whereas the datagram-aware scheduler ensures retrieval of around 75% of datagrams. We conclude that the tail-drop queuing policy (combined with a small buffer size) is indeed the cause of the undesired effect we observed.

## 4. CONCLUSION

In this paper we have presented results of the testing phase for the future EPFL smart-grid communication network infrastructure. When designing the experiments, we followed data transfer requirements described in the IEEE C37.118.2-2011 standard. However, we were able to identify critical scenarios where following these guidelines is not sufficient.

We used off-the-shelf SHDSL line terminals. When IP fragmentation occurs, two back-to-back packets arrive at the line terminal. When the reception rate exceeds the transmission rate, the packet queue is often full, and one or both incoming packets are dropped, thus rendering the reassembly impossible. In some of the considered scenarios, we find that packets arriving first are almost never lost, whereas a high loss-rate of the packets arriving second dictates the drop in goodput. We attribute this asymmetric behavior to the tail-drop queuing policy and small size of the queues implemented inside line terminals.

The resulting effect is that, even if the load is just slightly above the capacity of the SHDSL link, the goodput drops catastrophically. For example, when the offered load is 2.2Mbps, depending on the size of the packets, there are between 20% and 57% of lost Label B packets, although capacity is exceeded by only 10%. When capacity is exceeded by 20%, there are between 37% and 90% Label B packets that are lost. Thus, if measurement results are sent in fragmented packets, there is a risk that there is at least the same fraction of lost measurement results as the fraction of Label B packets that are lost. Hence, the impact on our monitoring system would be drastic.

There are two different approaches for mitigating the aforementioned issues. On one hand, a quick fix is to implement datagram-aware schedulers for the bottleneck queues. We show in Subsection 3.2 the benefits of this solution.

On the other hand, if we want to avoid losses altogether, PMUs/PDCs should implement traffic shaping on machines that run real-time operating systems (Real-Time Linux [4] for example). This would guarantee enough resources for processing and sending/forwarding packets within the desired boundaries, and packets would never be backlogged.

Therefore, a traffic management mechanism (currently out of scope of IEEE C37.118.2-2011 standard) is necessary if we want to ensure that networking resources are sufficient. A possible solution is the Integrated services architecture [5] that would guarantee quality of service. A sending device should go through a setup phase for resource reservation. During this phase all intermediate networking devices should accept or reject the reservation depending on available resources and the traffic specifications advertised by the sender, namely maximum packet size, peak rate, burst tolerance, and sustainable rate.

If we implement traffic shaping mechanism on devices with real-time operating systems with the integrated services network architecture, the system becomes much more deterministic and packet losses are avoided.

## 5. REFERENCES

[1] C37.118.2-2011 IEEE Standard for Synchrophasor Data Transfer for Power Systems. `http://standards.ieee.org/findstds/standard/C37.118.2-2011.html`.

[2] ZyXEL P-791R v2 product details. `http://www.zyxel.com/products_services/p_791r_v2.shtml?t=p`.

[3] ITU-T Recommendation G.991.2. Single-pair High-speed Digital Subscriber Line (SHDSL) Transceivers.

[4] Victor Yodaiken and Michael Barabanov. A real-time linux. *Linux Journal*, 34, 1997.

[5] Jean-Yves Le Boudec and Patrick Thiran. *Network calculus: a theory of deterministic queuing systems for the internet.* Springer-Verlag, Berlin, Heidelberg, 2001.