Impact of Packet Filtering on Time-Sensitive Networking Traffic

Lukas Wüsteney*‡, Michael Menth†, René Hummen‡, Tobias Heer*‡
* *University of Applied Sciences Esslingen, Germany, {lukas.wuesteney,tobias.heer}@hs-esslingen.de*
† *University of Tübingen, Germany, menth@uni-tuebingen.de*
‡ *Hirschmann Automation and Control GmbH, Germany, rene.hummen@belden.com*

***Abstract*—The Industrial Internet of Things, Industry 4.0 and cloud computing are fundamentally transforming today's industrial networks towards high connectivity. At the same time, the number of cyber-attacks against industrial infrastructure increased drastically over the last years, requiring to tightly limit the connectivity between the networked devices of a plant. For both of these trends, there are mechanisms evolving and partially already in place. Network segmentation with packet filters is a key mechanism for achieving improved network security while Time-Sensitive Networking (TSN) is a promising option to realize advanced real-time applications in future industrial networks. However, although being built based on widely accepted standards and despite their practical relevance, these two concepts don't play together well. In this paper, we analyze the problems that arise when TSN networks are segmented using today's firewalls and packet filters. In particular, we discuss and quantify the impact of delay and jitter caused by packet filters on TSN traffic. We also show that the delays and jitter introduced by CPU-based filtering can be prohibitively high in real-time scenarios. Based on our analysis, we present and compare three approaches to overcome the challenges created by the combination of these two major trends in industrial networks.**

***Index Terms*—Industry 4.0, TSN, Time-Sensitive Networking, Security, Network Security, Network Segmentation, IoT**

I. Introduction

Within the last years, industrial infrastructure has become the target of cyber security attacks more and more often. Recent attacks against industrial control networks led to damaged equipment, severe production outages as well as to irreversible loss of data. With the concept *zones and conduits*, the ability of an attacker to navigate through the network and to access data, functions and vulnerabilities of devices in the network is limited. This core concept of industrial network security is specified in widely accepted industrial norms and standards, such as the ISO/IEC 62443 [1] standard series. Figure 1 shows a simple example of the zones and conduits concept. The network is segmented into different zones (i.e., VLANs and subnets) and connected through conduits (i.e., restrictive filtering devices such as firewalls and switches with Access Control Lists (ACLs)).

Besides a push for better security, improving network performance has been a steady trend for industrial control networks. The introduction of Time-Sensitive Networking allows for fast and deterministic control communication based on standard Ethernet. TSN, as part of the IEEE 802.1 standards, introduces new mechanisms to deliver time critical traffic with

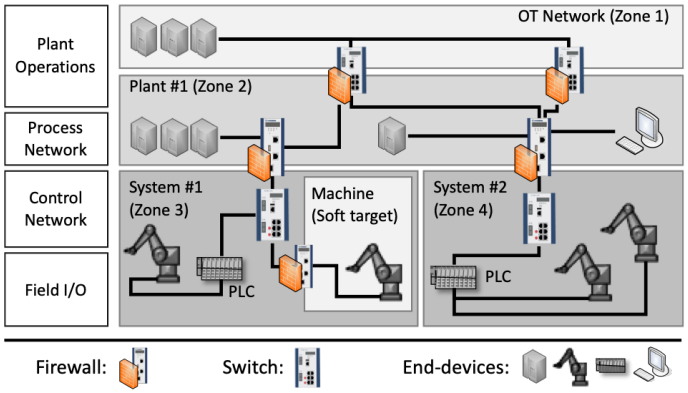


Fig. 1: Example visualization of industrial network segmentation

strict timing guarantees. TSN also supports converged traffic of different time criticality on the same wire. This enables industrial applications to use the same shared network for critical control traffic as well as for bulk data transmissions at the same tame. With TSN as future basis for industrial Ethernet, larger parts of the factory network can be used for in real time control traffic, and thus, new applications, such as control from the (local) cloud become feasible.

Looking at these two trends in isolation, the prospect of future industrial-control-networks could be fast, reliable and secure industrial control networks. However, properly implementing the concept of zones and conduits requires to segment a network into different and small zones. In practice, this translates into breaking down the network into smaller and smaller connected VLANs with filtering devices between these zones. TSN streams, however, are primarily designed to work within a single VLAN. The larger this VLAN becomes, the more devices can benefit from deterministic communication with other devices in the VLAN. Looking at these two trends it is obvious that there is potential for a conflict between security and performance. In this paper, we address this conflict and evaluate where new issues arise.

Our contribution is threefold: First, we analyze and evaluate the performance of two industrial packet filters. Second, we review the most important TSN specifications for conflicts with packet filters, assuming that these packet filters are not specifically designed for these protocols. In particular, we evaluate the impact of the forwarding delay of different types of packet filters on TSN traffic and explain the resulting limitations. Third, we discuss possible options for coping with

Accepted for publication in Proceedings of *17th IEEE International Conference on Factory Communication Systems (WFCS)*, Linz, Austria, June 9-11, 2021

the resulting problems. Our results can be the basis for the specification of TSN-capable packet filters and can enable existing security devices to process TSN traffic until such new packet filter are available.

II. Background

In this section, we briefly introduce the concept of *zones and conduits* and the mechanisms needed for its implementation. We also explain the generic system model of a packet filter and introduce typical traffic classes in industrial networks. Finally, we give a short introduction into the two TSN mechanisms "traffic scheduling" and "frame preemption".

A. Zones and Conduits in Industrial Networks

Because of its effectiveness, the implementation of zones based on Virtual LANs (VLANs) and conduits with packet filters is one of the prevalent security mechanisms in industrial networks. As such, this is a core concept of widely accepted general industrial security standards, such as the IEC 62443 [1], as well as sector-specific standards and regulations (e.g., NERC CIP for power distribution and transmission in North America [2]).

IEC 62443, Part 1-1 [3] Chapters 5.9, 6.5, and 5.10, introduce in the concepts of "Security zones" and "Conduits". The concept of network segmentation as described in Chapter 6, "Filtering/blocking/access control technologies", is highly relevant to TSN because it is a data plane mechanism as well. Logical or physical devices are grouped into separate zones with the help of security measures. Such zones can be segmented in sub-zones to restrict unnecessary access (compare Figure 1 "Zone 3" and "Soft target"). Through isolation, traffic is limited within zones while traffic across zone boundaries needs to be specifically allowed and filtered accordingly. There are two main mechanisms for implementing zones and conduits in industrial Ethernet: VLANs and packet filtering based on firewalls or ACLs.

1) Virtual Local Area Networks: VLANs define virtual subsets of LANs. The VLAN ID, defined in an additional header in the frame, associates a frame to a specific VLAN. Additionally, this 4 byte header stores a priority for the frame. In general, a frame cannot change the VLAN without a router. Switches forward a frame only within the specified VLAN and use the priority for the transmission selection. Multiple VLANs can share the same physical LAN.

2) Packet Filters: Conduits allow traffic to traverse the boundary of zones selectively. Typically, this is implemented with packet filtering by ACLs or firewalls.

ACLs filter traffic based on patterns in the headers of for example Ethernet frames or IP/TCP packets. This method of filtering is typically implemented in switching chips of networking devices (e.g., a switch, router, or firewall). Depending on the hardware support such mechanisms include for example accept or drop. ACL implementations differ drastically from vendor to vendor as they are a proprietary feature. The matching rules can either use explicit values or work with bit masks. In general, the matching capabilities of ACLs can be compared to stateless firewalls without deep packet inspection.

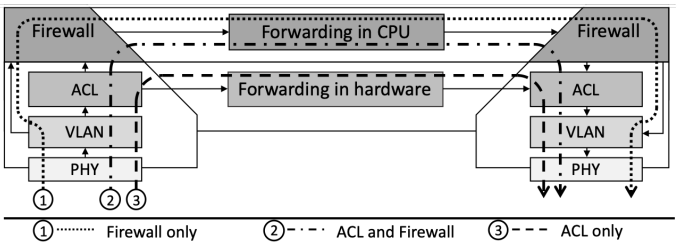


Fig. 2: Packet filtering device - system model

Firewalls connect IP subnetworks (e.g., different VLANs or different physical networks) and analyze traffic based on packet header or packet content. They process the rules defining permitted or prohibited traffic sequentially until the first rule matches. The rules cover the definition of addresses, address ranges, ports, upper-layer protocols (e.g., TCP), traffic direction, traffic volume and external factors (e.g., time of day). Firewalls are either stateless and stateful firewalls, depending on the ability to analyze and store the state of a connection (opening, open, closing, ...). In contrast to ACLs, firewalls are typically implemented for processing on CPUs. As part of the Linux kernel, *iptables* is the most common and a well-accepted implementation of a stateful firewall.

B. System Model of a Packet Filter

Figure 2 visualizes the system model of a packet filter. Some industrial firewalls do not have a switching chip in place and, therefore, cannot use fast ACL rules but resort to filtering in software (i.e., on the CPU, see path 1 in Figure 2). If the implementation has a switching chip with ACL capabilities packets typically will be processed by the ACL rules and the firewall rules (i.e., see path 2 in Figure 2). Some ACL implementations allow direct forwarding in hardware for single rules where packets are not processed by the firewall (see path 3 in Figure 2). As we show later, the capability to filter in hardware or software makes a significant difference when considering the forwarding of TSN traffic.

C. Time Criticality of Traffic Types

To assess the impact of the timing behavior of packet filters, we introduce two traffic types that are considered across multiple standardization groups. *Isochronous traffic* has cycle times below 2 ms (the lowest cycle time required is 31.25 µs) and does not tolerate any packet loss. The application (e.g., motion controller) is synchronized to the network and requires the packet to be received before a certain deadline. In comparison, *cyclic traffic* has cycle times of 2 ms to 20 ms and requires the latest arrival within a defined latency. Typically, process automation or cyclic safety-sensors operate on this level of precision.

D. Time-Sensitive Networking Mechanisms

TSN is a technology which enables time-critical applications to share the network with any other traffic by introducing mechanisms to provide determinism for the transmission. Such

shared networks, also known as converged networks, enable time-critical communication across long paths in the network. In IEEE 802.1 networks, the default transmission selection is "Strict Priority". This mechanism does not provide determinism. The forwarding mechanism selects frames residing in the egress buffer of a bridge/switch based on their traffic class. Second, the forwarding mechanism selects frames by their arrival time. However, traffic already in transmission on an egress port might block higher priority frames that arrived later. No deterministic communication is possible, as unpredictable interference creates jitter. The IEEE 802.1Q [4] standard defines to use the priority value in the VLAN header to determine a traffic class. To overcome the non-determinism and to enable converged networks, the IEEE 802.1 standardized the two following transmission selection algorithms.

1) Time-aware Traffic Scheduling: IEEE 802.1Qbv-2015 [5] or "Enhancements for Scheduled Traffic" standardizes the time-aware traffic scheduling and cyclic schedules for the forwarding of specific traffic classes. The default transmission selection between all active traffic classes is strict priority. The configuration of the time-aware scheduler ensures that no lower-priority traffic in transit blocks the time-critical traffic. This standard is part of IEEE 802.1Q-2018 [4] and mainly defined in Sections 8.6.9, 8.6.10, 12.29.1, Annex B.15, Annex Q and Annex S.

2) Frame Preemption: In any transmission selection algorithm defined by the IEEE 802.1, a frame already in transmission blocks any other frame, without considering priorities or time-aware configuration. Hence, low-priority traffic can delay frames with a high priority. The frame preemption standards IEEE 802.1Qbu-2016 [6] and IEEE 802.3br [7] specify the interruption of preemptable frames in transmission by express frames. The priorities belonging to the express category is a subset of all 8 VLAN priorities. All remaining priorities belong to the preemptable category. This mechanism helps to reduce the worst-case time a high priority frames waits at each hop (i.e., at each bridge/switch) in the network. In the Sections 8.6.8, 12.30.1 Annex B.16, Annex R and Annex S, the IEEE 802.1Q-2018 [4] standardizes the frame preemption.

III. Evaluation of Segmentation Mechanisms in Relation to Industrial Automation Traffic

Packet filters are the core building block for zones and conduits. Despite the prevalence of the zones and conduits concept and the rise of TSN, to the best of our knowledge, the effect of different packet filters on TSN traffic has not been discussed in literature. The assumption, that firewalls must be compliant with TSN mechanisms or that TSN traffic must be strictly limited to a single subnetwork, is intuitive. However, to the best of our knowledge, there are no TSN-capable firewalls (neither as products nor as concepts) as of today. With concepts, such as control from the cloud, controller to controller, or computing in the (on-premise) cloud, time critical communication traverses a growing number of (sub)networks or zones. Due to the risks associated with enlarging security zones, the chosen security mechanisms need to be compatible

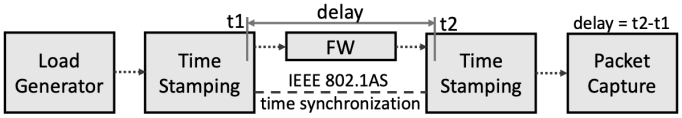


Fig. 3: Measurement setup. Time-stamped TSN traffic is used for measuring delay/jitter.

with the necessary Quality of Service (QoS) requirements, to keep the security zones small. Modern industrial protocols, like OPC UA PubSub, enable communication from the cloud and local cloud, as well as from controller to controller.

Using TSN across zone boundaries raises new challenges for on-path packet filters. We first introduce our methodology and the measured packet filters. Second, we measure the performance, specifically the timing, and generic characteristics of packet filters. Later we discuss on a conceptual level and quantify the impact of the delay and jitter introduced by different filtering approaches on TSN traffic.

A. Methodology

Our goal is to show generic characteristics regarding the timing behavior of packet filter implementations. Figure 3 visualizes our measurement setup. We implement the synchronization between the two time-stamping units with the IEEE 802.1AS protocol and achieve a precision of 30 ns. We argue that this precision is sufficient to show the generic characteristics for two reasons: a) all of our results show a delay and jitter at least tenfold higher than the synchronization precision and b) TSN also relies on the precision of IEEE 802.1AS for the planning of schedules. As we present in our evaluation and discussion, the exact numbers are less important, than the identified characteristics under certain configuration and load scenarios. The *time stamping* units visualized in Figure 3 insert their current time into the frame payload upon transmission and reception. The delays presented in this work refer to the difference between these two time stamps and are calculated offline after capturing all packets.

The measurements we performed cover a wide range of different packet sizes and packet rates. However, to keep the presentation concise, we focus on the results for 512-byte UDP packets with varying packet rates. The packet size we chose is a compromise between short packets, typical for industrial control applications, and large best effort traffic such as video streams. For analyzing packet filters, the variation of the packet rate is important, as this directly influences the number of operations done by a packet filter. In our diagrams we always present all measurements in absolute numbers without erasing outliers. We present the maximum and minimum values with the upper and lower whiskers in the boxplots. The boxes signify the distribution, representing the upper and lower quartile of all measured delays.

We measure the forwarding behavior of two industrial firewalls: The Eagle30 industrial firewall (FW1) as well as the Eagle40 industrial firewall (FW2) by Hirschmann Automation and Control GmbH. The Eagle30 is a firewall with a 667 MHz CPU that supports hardware ACLs based on the Broadcom Hurricane switching chip. The forwarding is either possible in hardware or via the CPU, as shown in Figure 2 with

	FW1 (Eagle30)	FW2 (Eagle40)
CPU	0.667 GHz	1.33 GHz
Switching Chip	Broadcom Hurricane	n.a.
ACL Rules	176	n.a.
Firewall Rules	2048	2048
Link Speed	1 GBit/s	1 GBit/s

TABLE I: Summary of FW1 and FW2 specifications.

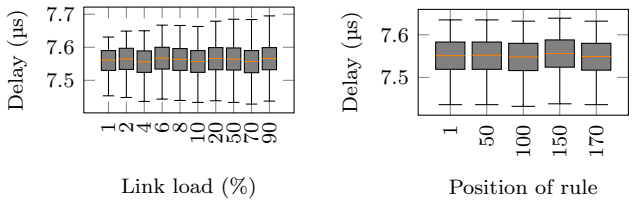
the paths 2 and 3. The Eagle40 is a firewall with faster 1.33 GHz CPU. However, the Eagle40 has no switching chip and therefore no support for ACL based filtering (only supports path 1 visualized in Figure 2). We chose these two types of firewalls since they show the difference between hardware-based and software-based filtering. Table I summarizes the hardware details. FW1 implements the complete system model visualized in Figure 2, whereas FW2 does not implement the ACL filtering and hardware-based forwarding. Both devices use the *iptables* firewall implementation which is executed on the general-purpose CPU of the firewalls. We emphasize that we do not intend to give a comprehensive performance analysis of these firewalls but use these two commercial firewalls and their results to illustrate specific areas of interest. Similar behavior is expected for other firewalls from other vendors, depending on whether or not they rely on CPU-based or hardware-based ACL filtering. Although it seems obvious that ACL filtering is faster than software filtering, we still quantify the results and analyze the latency characteristics of both, to show the compatibility with TSN mechanisms.

B. Timing Performance of Packet Filters

Packet filters inspect the content of a packet and forward or drop the packet based on a ruleset. Different types of packet filters inspect either just the header fields or the headers as well as the payload of the packet. The latter one is the case when the packet filter performs Deep Packet Inspection, which is not analyzed in this work. All packets that leave a zone (VLAN) are subject to inspection.

Packet filters are either implemented in hardware (e.g., processing ACL rules by the switching ASIC of a switch) or in software (e.g., processing of the filter by the CPU of a firewall). For time-critical traffic, the duration of this inspection matters since it delays the forwarding of the packet by a fixed or variable amount of time. The difference between hardware and software implementations is the delay caused by the general processing time. With additional load on the system, hardware implementations operate constant and cause no additional jitter. Packet filters that rely on software-based filtering on a CPU are far more affected by the load of the system. In this chapter we provide measurements of the impact of hardware-based and software-based filtering on the delay and jitter of TSN traffic.

1) ACL Delay and Jitter: We first measure the forwarding delay of filtering based on ACLs with different load scenarios. Figure 4a shows a constant forwarding delay for ACL rules on FW1 in the order of 7.6 µs. The whiskers signify the minimum and maximum delay, indicating the expected range for the arrival of the frame on the switch after the packet filter. The figure also shows that the results are independent of the



(a) Delays for different link loads (matching rule: 100)

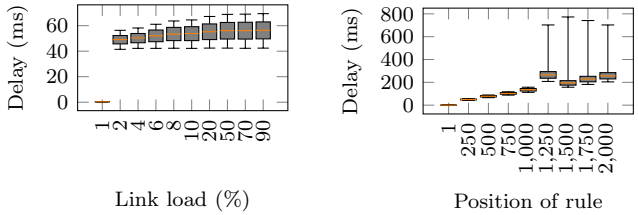
(b) Delays for different matching positions (link load: 2%)

Fig. 4: FW1 delays for filtering in hardware (Packet size 512 Byte; Duration: 10 s)

network load and have a symmetric jitter. We use *jitter* as the maximum difference between the average and the minimum and maximum values. Whereas *spread* or *range* are defined as the complete window between the two whiskers. Symmetric jitter describes that the positive, as well as the negative, jitter form the average delay are the same. The jitter, which is smaller than 0.2 µs, is magnitudes smaller than the total delay.

While offering constant and fast performance, hardware filtering implementations are not as flexible as CPU-based filtering, and hence, are limited in their complexity. Therefore, ACL implementations usually only allow a limited number of rules. FW1 can process up to 176 ACL rules per interface. We repeated our measurements with different positions for the matching rule to determine if a later matching in the rule processing affects the filtering performance. Figure 4b shows that the forwarding delay for different rule positions is constant. Packets matched by later rules in the rule set do not create a penalty in form of additional delay. Overall, these measurements show that ACLs only introduce a negligible processing delay and range of delay, making these filters well suited for high demanding industrial traffic classes.

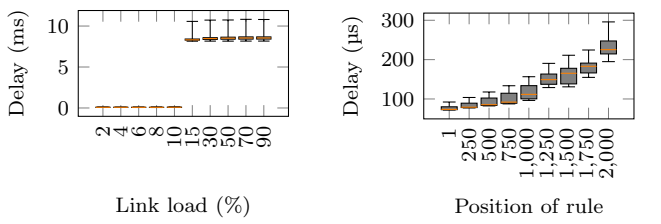
2) Firewall Delay and Jitter: Software implementations are flexible regarding content of inspection and number of rules. CPU-based processing has the downside of larger jitter introduced by the load on the system since the CPU is also used for other purposes at the same time. FW1 and FW2 have a maximum of 2048 firewall rules per routing interface. Figure 5a shows the forwarding delay for FW1 and Figure 6a for FW2 with software filtering under varying load conditions. The firewall delays of FW1 are multiple orders of magnitude higher than for ACLs. Each measurement on its own, with a known and fixed load setting, introduces a maximum jitter of 13 ms. Whereas an unknown load situation results in a delay range of 69 ms, as the total minimum and maximum across all measurements define the worst-case delay range. Considering the low delay and jitter requirements of TSN, FW1 introduces



(a) Delays for different link loads (matching rule: 250)

(b) Delays for different matching positions (link load: 2%)

Fig. 5: FW1 delays for filtering in software (Packet size 512 Byte; Duration: 10 s)



(a) Delays for different link loads (matching rule: 250)

(b) Delays for different matching positions (link load: 2%)

Fig. 6: FW2 delays for filtering in software (Packet size 512 Byte; Duration: 10 s)

significant delays even for low link utilizations (starting from 2% link load (20 Mbit/s)). FW2 introduces a significantly lower delay. However, with increasing network load (between 10% and 15% link load (100-150 Mbit/s)) a delay of 10 ms with a jitter of 2.2 ms is to be expected. Compared to typical cycle times in TSN (e.g., 100 µs to 1 ms), a delay of 10 ms and a jitter of 2.2 ms is prohibitively large, since all packets are heavily desynchronized with respect to the cycle after traversing the CPU-based filter of the firewall.

A later position of the matching firewall rule in the rule set results in more operations executed by the CPU. Therefore, we repeated our measurements to further evaluate the impact of the position on both firewalls and the associated CPU operations. Figure 5b shows that the CPU linearly processes the firewall rules on FW1. Later rules require more CPU cycles and introduce more time for the processing. With a higher number of processing steps, the jitter increases non-symmetric and unpredictable. This asymmetric jitter creates a forwarding behavior, which is complex to model. Different packets forwarded by the firewall may be delayed differently solely based on the arrangement of the rules of the firewall. Not all packets of a stream may match the same rule, as rules also depend on properties of a packet. Together with configuration changes during runtime, this leads to varying delays in a running system. Again, Figure 6b shows that the faster CPU of FW2 leads to a more predictable behavior. The non-symmetric jitter of 60 µs for packets that are processed at the end of the ruleset is in conflict with cycle times for synchronized motion applications (e.g., 100 µs).

IV. Impact of Delay and Jitter on TSN Traffic

The data-plane mechanisms of TSN consist of transmission selection algorithms. These provide forwarding guarantees (i.e., upper bounds on latency, bandwidth, and jitter) to packets or data streams in an Ethernet network. Deterministic communication in industrial networks uses either one or multiple TSN standards in combination. In this section, we first introduce expected failures for misbehaving devices in a TSN network. Later we compare the impact of the previously presented behavior of packet filters to the requirements of industrial applications and to the TSN mechanisms.

A. TSN Application Failures

In this section, we classify the effects of delays and jitter on the network and end-devices to determine the criticality of these for applications.

1) Application Failures due to Late Frames: Jitter creates a range for the expected arrival time of the frame. If the transmission delay is larger than the defined application-requirements, the packet delay is unacceptable for the end-device and the application will fail (e.g., transition to a fail-safe state without proper function or cause harm). Depending on the TSN mechanisms used in combination with the packet filters, delay can be static (all frames arrive late) or dynamic (some frames arrive late, sporadic failures).

2) Cross Application Failures: Even if a slightly delayed frame arrives in time at the end-system, this delayed frame can still have a negative effect on the frames of other streams. This can happen by filling up queues of forwarding devices or by blocking high-priority time slots at the switches that are needed for the transmission of other frames. Hence, the application that uses the slightly delayed frame will not fail but the traffic of other streams becomes non-deterministic. Such a conflict between two or more frames of the same priority can be caused by one or more of the following conditions:

- A late frame arrives just before the frame that is supposed to be transmitted in a time frame and takes the bandwidth of the second frame, causing the second frame to be delayed as well.
- A late express frame preempts a preemptable frame, which can only be preempted by a second express frame after the next fragment, causing a small, but unplanned delay of this express frame.
- A late frame causes the remaining time in a time slot to be smaller than a second frame needs for transmission. The second frame needs to wait for the next time slot for its according traffic class.

On single hops, these conflicts introduce just small additional delays for other frames. However, these non-deterministic delays add up throughout the network. Even worse, a data-stream crossing a packet filter can influence any other intersecting stream. Figure 7 visualizes this effect. The data-streams 1 and 2 are transmitted as planned on the links A and C. After the packet filter, data-stream 1 is unsynchronized on link B. These unsynchronized packets cause data-stream 2 to be delayed on link D as well. The unsynchronized data-stream 2 can influence any other data-stream in the network itself. As a result, other applications may fail, even if these do not communicate across the non-conforming packet filter.

B. Influence of Firewall Performance on TSN Guarantees

QoS guarantees in TSN networks are based on bounded latency and fixed arrival intervals of frames. The stronger the requirements and the better the guarantees are, the smaller the

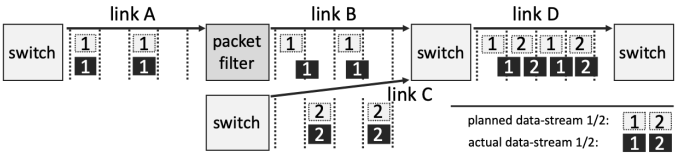


Fig. 7: Cross application influence of delays. The figure shows packets delayed by the packet filter as well as packets delayed by other delayed packets.

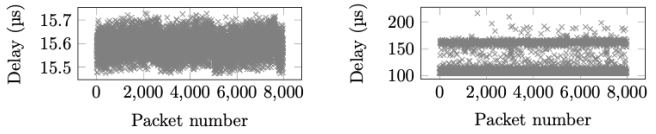
spread of the arrival time. Larger jitter on the transmission path results in more bandwidth allocated to guarantee the QoS requirements. Within this section we show the direct influence of the measured packet filters and typical industrial requirements on the allocated bandwidth.

Figure 8a and 8b show the distribution of delay over time for the CPU-based firewall processing on FW2 and the ACL-based firewall processing of FW1, respectively. FW1, with its ACL-based filter, shows a much more predictable delay behavior in a range that is suitable for time-critical applications. For FW2, challenging time-critical applications may experience desynchronized and late packets. Both figures visualize delay differences in 2 ways: a) A static delay may be undesirable, but it can be dealt with at the time the schedules for TSN are calculated. b) A large jitter and delay range is more problematic since it causes variable delays at runtime.

Depending on the TSN mechanisms in the different security zones, these results have different impact on the compatibility of filtering devices and TSN. For TSN networks with traffic scheduling the traffic can be scheduled within the network per class or per stream. We assume end-devices following a predefined schedule for transmitting their data. For frame preemption, end-devices transmit all streams based on a schedule, but the network is unaware of this schedule. Within the following three sections we show and calculate the bandwidth overprovisioning per TSN mechanism.

1) Class-based Traffic Scheduling: Opening the transmission window for a complete class of frames at once, means that the transmission of all frames is back-to-back. Any slightly delayed frame results in delaying the following frames of this class. All frames delayed by more than the packet length of the next planned frame causes reordering and delays of later frames. There are two strategies for planning with increased delay-ranges in class-based scheduled networks: a) If the delay range is smaller than the cycle time, the slot duration can be enlarged by the worst-case delay range of all frames. This ensures, that no frame misses the slot and needs to wait for the next cycle. b) If the delay range is larger than the cycle time, the delay range defines in how many cycles the frame could possibly arrive. The transmission of the frame needs to be possible in any of these cycles.

To visualize strategy a), introduced above, we assume an isochronous application with a cycle time of 2 ms. The packets inspected by FW2 match rule position number 2,000 and have a jitter of 60 µs and a delay range of 100 µs at 2% link load (compare to Figure 6b). On a 1 GBit/s link, a frame of size 512 bytes takes around 4.5 µs per hop in store-and-forward transmission. For 100 streams in this scenario the transmission



(a) Delay for hardware filtering on FW1 (matching rule: 100)

(b) Delay for software filtering on FW2 (matching rule: 250)

Fig. 8: Delay distribution for ACL and FW filtering; Packet size (1024 Bytes) and load on the link (1%) is constant; Duration: 10 s

takes around 450 µs. Adding the worst-case delay range, as described above, the slot time increases about 25% to around 550 µs. As example for strategy b) we assume a 10 ms cycle time for cyclic traffic and 2% link load on FW1. The frames match firewall rule number 1,000, resulting in a delay range of 40 ms (see Figure 5b). The expected arrival is four times the cycle time and four times the bandwidth needs to be reserved.

2) Stream-based Traffic Scheduling: In comparison to class-based scheduling, stream-based scheduling opens the transmission gate per stream. Any frame missing its slot will hijack the slot of the next stream in the same class, because the transmission gate is related to the class of a frame. To prevent this negative effect, the slots need to be enlarged in relation with the increasing jitter and delay range.

As a first example, we compare the optimal transmission time of one 512 byte packet on a 1 GBit/s link of 4.5 µs to the influence of FW2. On FW2 a packet matching position 500 with 2% link load has a delay range of 35 µs (compare to Figure 6b). This results in an increase of the original slot time (about 5 µs) of factor 8 to 40 µs. In comparison ACL filtering on FW1 introduces a maximal delay range of 0.3 µs for all presented measurements (see Figures 4a and 4b). As second example the ACL filtering results an increase of the slot duration about less than 10% to 5.3 µs.

3) Frame Preemption: Without knowledge about preemptable traffic, the worst-case delay per hop is about 1 µs for express frames. This increases the delay range per hop due to unplanned frame preemption of preemptable traffic. In order to meet the requirements, end-devices need to schedule the transmission, such that express frames do not interfere with each other in the network. This means, that delay ranges of all express frames should not overlay on any hop in the network.

As first example we evaluate packets matching firewall rule position 2,000 on FW2. A 512 byte packet in this scenario has a delay range of 100 µs. This results in a distribution of high priority frames with a gap of 101 µs. Assuming a network cycle of 10 ms, this means that less than 100 streams are possible within the network to guarantee optimal forwarding. In comparison, ACLs introduce a delay range below 1 µs and result in a transmission gap of less than 2 µs. This second example shows, that theoretically more than 65,000 streams are possible with frame preemption and ACL packet filtering.

Summarizing all three scenarios and their examples results in the conclusion that jitter is the root cause for more bandwidth allocation or weaker guarantees to time critical traffic. All overprovisioned bandwidth is fixed for all following hops.

V. Solutions to Limit the Influence of Jitter

Based on our evaluation, we discuss three different approaches for combining TSN with a packet filter. First, we introduce how planning of streams in the network can deal with jitter introduced by packet filters. Second, buffering frames before further transmission helps to reduce jitter and unused bandwidth. And last, we discuss the usage of ACL-based filtering for time critical traffic.

A. Considering Filtering Delays in TSN Scheduling

The first approach to deal with the delays of firewalls is to estimate the delay and the jitter. TSN networks have a planning entity to ensure that time-critical traffic does not interfere with each other. If the delay and jitter is known and can be estimated per packet filter, the planning entity can consider this information for the stream calculations. The solution of the stream configuration needs to be based on delay and jitter values that are either static over the runtime of the network or dynamic and therefore defined by a range of values for all possible occurring situations.

As of today, there are no estimation schemes to determine firewall delays in the time scales necessary for TSN. The measurements we perform only vary in single dimensions (e.g., load, or matching position) to present important characteristics. But in operative networks many parameters can vary at the same time. Even if the planning entity could estimate filtering delays for the current packet filter configuration, these delays would not be static over the runtime of a network and thus not fit to the static configuration of TSN.

For example, Figures 5b and 6b show the impact of the matching position on the filtering delay. The position of the matching rule may depend on various packets and environment properties and might therefore be difficult to predict. Moreover, any change in the firewall configuration or the firewall state could result in a non-trivial and non-linear change in processing time. Any new or deleted rule causes the complete ruleset to be shifted by one position, but the position a specific packet matches shifted to an arbitrary different position in the ruleset. Also, changes in non-critical-traffic-rules influence the overall performance, as all rules are processed on a shared CPU. Hence, with each change of a firewall ruleset all existing schedules need to be recalculated and reconfigured.

B. Using Frame Buffering to Restore Synchronization

Instead of trying to find small and precise estimates for the firewall processing time, the planning entity can use worst-case assumptions for the processing delay and jitter (e.g., based on the maximum number of rules and the maximum load). As our measurements show, this results in high delays and large delay ranges compared to the best-case and average performance. However, these large values are constant over the runtime of the network. On the downside, this leads to high bandwidth consumption, higher end-to-end delays and the requirement to artificially delay all frames towards the worst-case to maintain the synchronization. In our tests, these worst-case assumptions would amount to 8.5 ms processing delay and 2.5 ms jitter for FW2 and 65 ms delay with 24 ms jitter on FW1. Assuming a cycle time of 20 ms, FW2 requires more than 10% of the cycle time to be allocated on the path behind the firewall to accommodate the jitter for FW2. On FW1 the jitter is even higher than the cycle time. For ACL based filtering, the planning with worst-case values is much better because of the constant and low delays and jitter (7.55 µs / 0.25 µs).

One version of this approach is standardized as "Cyclic Queueing and Forwarding" (CQF) in IEEE 802.1Qch-2017 [8]. As part of IEEE 802.1Q-2018 [4] it defines in the informative Annex T, a specific use case of the traffic scheduling mechanisms. The basic idea is to store all frames arriving within one cycle until the next cycle and to transmit the frames in bursts. At first sight this generates an even larger delay per hop, as frames are buffered at every CQF node. However, it allows calculating accurate estimations of the forwarding delay since it eliminates jitter caused by interfering traffic. Hence, it serves to create determinism at the cost of an increased overall latency and is placed on the stream path only directly behind packet filters.

This mechanism prevents desynchronization in two different cases: a) If the delay range is smaller than the cycle time and b) if the delay range is larger than the cycle time. In case a), in which the range is smaller than the cycle time, late frames are buffered until the next cycle. Regardless of their original delay, these frames are forwarded at a defined time in the next cycle. This reduces the overall jitter at the cost of a fixed delay. However, a fixed delay and jitter can easily be dealt with in the network planning phase, improving the guarantees and reducing allocated bandwidth. In the second case b), the range is larger than the cycle time. In this case the delay range will not significantly be reduced by CQF. Yet, this mode has a positive effect: the frames are released at a specified time (i.e., multiples of the cycle time), which reduces the random displacement of other frames in the network. However, such mechanisms increase in both scenarios the average delay and require large buffers in network devices.

C. Combining ACLs and Software Filtering

Our measurements show that ACL based filtering of packets is well suited for TSN traffic. The delay and delay range introduced by ACL filtering is constant over all measurements. However, using ACLs also has disadvantages: Devices with ACLs generally support fewer rules as software firewalls, their implementation is vendor specific and the rules are less flexible (e.g., no stateful filtering) than rules for CPU based filtering. Moreover, ACLs are not well suited for routing between subnetworks so that zones may not always match to individual subnetworks. A combination of both approaches (ACLs and CPU based filtering) can provide performance for few time-critical streams and flexibility for other transmissions. Based on the ACLs of FW1 we configure the fast-path through the switching chip, avoiding the CPU-based filtering (see path 3 in Figure 2) for TSN traffic. This reduces the additional filtering delay for critical traffic to the constant delay and low jitter of the ACLs. The ACL based filtering delay range of FW1 is in the order of 0.2 µs (see Figure 8a). This jitter increases the processing time by 1% compared to the average delay.

Combining the planning and buffering approach together with the use of ACL filtering, the jitter is even less than 1% of the smallest cycle time requirements of isochronous traffic (31.25 µs). Knowing this constant and small delay range, a TSN planning entity could shift the opening of the time aware gates slightly along the path behind the filter. Frames will not be buffered for a complete cycle, but just to be aligned on

the worst-case delay. Therefore, ACL based filtering is even suited for the most challenging motion applications.

VI. Related Work

Firewalls are prevalent in most industrial networks but as of today they are rarely used for time-critical communication. As a consequence, developers and researchers often focus on evaluating and optimizing throughput as major performance indicator. However delay and jitter are two additional critical performance indicators for packet filters in TSN networks. In this section, we present related work that discusses the timing behavior of firewalls as well as generic TSN scheduling strategies to cope with delay and jitter.

Zvabva et al. measured [9] the influences of firewalls on industrial communication for the Modbus/TCP protocol. Besides the fact of jitter depending on the position of the rules, the authors also present the influence of additional security features like Deep Packet Inspection. The analysis, however, only focuses on a maximum of 18 rules with low and constant traffic and does not consider TSN and its substandards.

In [10] Cheminod et al. analyzed the impact of cross traffic on industrial firewalls. The authors perform the analysis by evaluating Modbus/TCP in different settings with different load scenarios. The same authors also implemented a protocol independent evaluation in [11]. TSN mechanisms and the challenging timing requirements of TSN traffic are not considered.

Within [12] Stylianopoulos et al. present the difference of software frameworks on packet forwarding. However, this work does not analyze the strict sequential execution of filtering and therefore additional internal blocking of frames.

Hellmanns et al. [13] present an approach to handle small timing imprecisions of end-devices in the planning phase of TSN networks and schedules. Their work focuses on delay and jitter caused by end systems but does not consider filtering devices nor network segmentation.

Hasan et al. [14] propose a latency-aware segmentation of networks into isolated zones to satisfy the timing requirements of time critical applications. However, current trends in industrial networks propose field-level to cloud communications and control loops that connect large parts of a site. The proposed small and latency aware zones are not capable of covering the long communication paths between cloud and field network.

Heimgaertner et al. [15] introduce a mechanism to bypass a firewall for known connections with SDN switches to reduce the load of the firewall. The authors assume non time-critical IT traffic and focus on throughput instead of delay and jitter.

VII. Conclusion

In this work, we analyzed the compatibility of the *zones and conduit* concept, specifically packet filtering mechanisms, in industrial networks with TSN. We show that compatibility of the packet filtering with real-time TSN mainly depends on the delay and jitter introduced by the filtering device. This leads to different symptoms ranging from late frames to desynchronized packets, which can delay other time-critical transmissions. We analyzed and described the influence on such applications communicating across such packet filters and on such just communicating in the same network.

Our evaluation of two industrial firewalls quantifies the delays and jitter to show their relation to the processing guarantees that real-time TSN traffic requires. Based on the results, we can conclude that packet filtering and forwarding on general-purpose CPUs is not deterministic and difficult to predict for dynamic network loads. On the contrary, we show that hardware assisted ACL-based filtering only introduces minor delays. Later we analyzed three different deployment scenarios, which all seek to overcome the introduced effects and challenges. In combination, we achieve packet filtering which is compatible with TSN and isochronous applications with cycle times of 31.25 µs.

In future TSN deployments, TSN will be used in fully automated deployments (i.e., schedules are automatically generated as new devices join the network), future network scheduling mechanisms must become aware of the complex timing behavior of filtering devices. This would enable to design networks with small zones for security reasons while maintaining the ability to transmit real-time TSN traffic across the zone boundaries.

References

[1] International Electrotechnical Commission (IEC), "IEC 62443: Industrial communication networks – Network and system security."

[2] North American Reliability Corporation, NERC), "ICIP-005-5, Cyber Security – Electronic Security Perimeter(s)."

[3] International Electrotechnical Commission (IEC), "IEC 62443-1-1: Terminology, concepts and models," *IEC/TR 62443*, 2009.

[4] "IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks," *IEEE Std 802.1Q-2018*, 2018.

[5] "Amendment 25: Enhancements for Scheduled Traffic," *IEEE Std 802.1Qbv-2015*, 2016.

[6] "Amendment 26: Frame Preemption," *IEEE Std 802.1Qbu-2016*, 2016.

[7] "Amendment 5: Specification and Management Parameters for Interspersing Express Traffic," *IEEE Std 802.3br-2016*, 2016.

[8] "Amendment 29: Cyclic Queuing and Forwarding," *IEEE 802.1Qch-2017*, 2017.

[9] D. Zvabva, P. Zavarsky, S. Butakov, and J. Luswata, "Evaluation of Industrial Firewall Performance Issues in Automation and Control Networks," *29th Biennial Symposium on Communications, BSC 2018*.

[10] M. Cheminod, L. Durante, A. Valenzano, and C. Zunino, "Performance impact of commercial industrial firewalls on networked control systems," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2016.

[11] M. Cheminod, L. Durante, L. Seno, and A. Valenzano, "Performance Evaluation and Modeling of an Industrial Application-Layer Firewall," *IEEE Transactions on Industrial Informatics*, pp. 2159–2170, 2018.

[12] C. Stylianopoulos, M. Almgren, O. Landsiedel, M. Papatriantafilou, T. Neish, L. Gillander, B. Johansson, and S. Bonnier, "On the performance of commodity hardware for low latency and low jitter packet processing," in *Proceedings of the 14th ACM International Conference on Distributed and Event-Based Systems*. New York, NY, USA: Association for Computing Machinery, 2020, p. 177–182.

[13] D. Hellmanns, J. Falk, A. Glavackij, R. Hummen, S. Kehrer, and F. Dürr, "On the performance of stream-based, class-based time-aware shaping and frame preemption in TSN," *IEEE International Conference on Industrial Technology*, pp. 298–303, 2020.

[14] M. M. Hasan and H. T. Mouftah, "Latency-aware segmentation and trust system placement in smart grid SCADA networks," *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD*, pp. 37–42, 2016.

[15] F. Heimgaertner, M. Schmidt, D. Morgenstern, and M. Menth, "A software-defined firewall bypass for congestion offloading," *13th International Conference on Network and Service Management, CNSM 2017*.