"4I +1/2P + [P+SFD] +EFD + 7I"

or

"How to gain 3.2 ns with Corrupted Speculative Triggering (CST) without being detected, and increase by 3 times EUREX Order Entry connectivity revenues" *.

MOSAIC FINANCE, July 1st, 2025

* This document does not aim to promote the irregular practice of CST. Conversely, it is undertaken in good faith and in an effort to promote transparency regarding the fraudulent practice of CST.

Abstract

We reveal in this paper the detailed technical implementation of a corrupted version of the Speculative Triggering practice that we believe to be performed on EUREX.

The format we disclose is only one version of the many possible formats of CST, widely implemented by some Ultra Low Latency (ULL) participants.

Any CST format consists in a shortened Ethernet packet sent at a frequency of tens of Millions per second, which violates several clauses of the Ethernet standard and of EUREX Network Access Guide. CST is intentionally structured in order to exploit loopholes in EUREX monitoring procedures and remain undetected by the counters EUREX is currently monitoring.

Thanks to the 3.2 ns latency advantage of orders built with this technique, a negative reaction time to the reception of a Market Data can be achieved and priority is gained on market participants behaving orderly.

We secondly provide in this document a detailed and straightforward procedure to evidence in real time the CST practice using EUREX's current monitoring equipment ARISTA.

We thirdly evidence that in order to fully benefit from CST's latency advantage, participants performing it must subscribe three times as many 10Gbit/s Order Entry connections as would be necessary when behaving orderly.

The CST practice thus increases EUREX connectivity revenues by tens of millions of euros per year.

A potential conflict of interest is brought up by tolerance for a practice that so clearly contravenes EUREX's own policies.

"Regular" Speculative Triggering

Deutsche Börse has incentivized so called "Speculative Triggering", allowing market participants to issue Ethernet packets as soon as they receive the start of a Market Data packet.
Speculative Triggering enables participants to adapt dynamically the contents and the ultimate destination of the packet in construction, while "decoding" the content of the incoming Market Data packet. All packets created in the Speculative triggering process can be converted into actual market orders, but only a fraction of them are ultimately converted into orders, most of these packets being diverted to the so-called "Discard IP".

Speculative Triggering in itself does not raise regulatory issue, as Eurex has documented and defined boundaries to the practice (see below "Document 1").

Details of the Speculative Triggering practice are given by Deutsche Börse, page 8 to 10 of the following document:

https://www.eurex.com/resource/blob/48918/4f5fd0386f272f89219bd66c0a546d09/data/presentation_insights-into-trading-system-dynamics_en.pdf (Document 1)

Deutsche Börse has set limits in the "Speculative Triggering" practice in two ways in its Network Access Guide (NAG):

> **Qualitatively**, the format of the messages market participant are allowed to send must comply with, among others, §12.1.1 of the NAG:
>
> "*Use network protocols in an orderly fashion, that do not for example systematically:*
> *– send* ***corrupted ethernet frames*** *(e.g.* ***shortened ethernet packets****, padded packets for packet length > 64 bytes) (…)"*
>
> and § 12.1.2 of the NAG which prohibits to:
> - *"Send packets not reaching the gateway."*
>
> **Quantitatively**, a limit of 30 000 Ethernet frames per second and 600 000 Ethernet frames per minute.
>
> *"Those limits include* ***all Ethernet frames irrespective of content*** *and whether they are sent to the discard IP addresses or not."*

References to these limitations can be found page 122 and 123 of the Deutsche Börse Network Access Guide:

https://www.eurex.com/resource/blob/4417688/d534a09c9701dfb39015f456a4c46402/data/N7_-_Network_Access_Guide.v.2.6.pdf (Document 2)

Technically, Speculative Triggering dynamic can be summarized in the following way:

To minimize the latency between the publication of an order book update and the subsequent order, participants detect the start of the market data packet and immediately start sending an outgoing Ethernet packet.

As quoted by Deutsche Börse page 10 of Document 1: "*The fastest participants may react and send a response* ***as early as*** *the first bit of market data has been received, dynamically reading the market data packet while already streaming out the response..*"

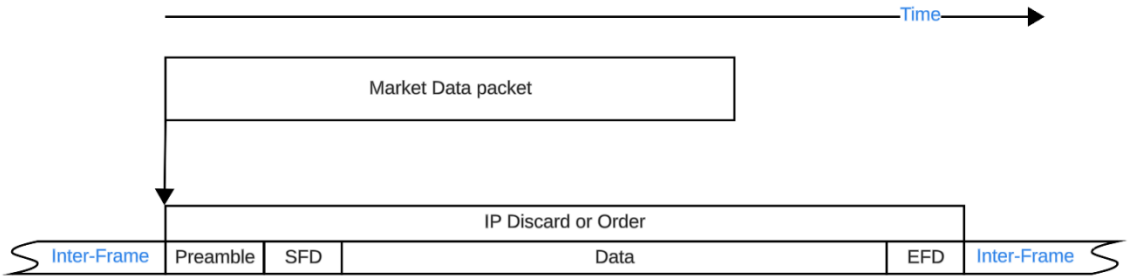
This outgoing Ethernet packet is, at the very beginning, an IP Discard (see § 12.3 of Document 2) and is dynamically transformed during its construction into an Order in case the market data packet is appealing (if not it remains an IP Discard that will be rejected by the Access Layer Switch of the Deutsche Börse network).

An IP discard (or an order or any type of Ethernet packet) **which complies with Ethernet protocol** (see IEEE 802.3, Clause 46.2, standard for Ethernet) has the following structure:

[P + SFD + Data + EFD], where

- P is the Preamble, which is a set of 7 Bytes
- SFD (Start of Frame Delimiter), which is a 1 Byte delimiter to declare the start of the data stream
- Data: is the Ethernet Frame (with a minimum required size of 64 Bytes)
- EFD (End of Frame Delimiter) is a 1 Byte delimiter to declare the end of the data stream

A conformant Ethernet packet thus has a minimum length of 72 bytes. An Inter Frame gap is also required between two frames.



Considering the qualitative and quantitative limitations imposed by Deutsche Börse NAG, it is impossible for a market participant to systematically start creating an outgoing message **before** the actual reception of the beginning of an incoming market data packet (see mathematical proof in Annex IV).

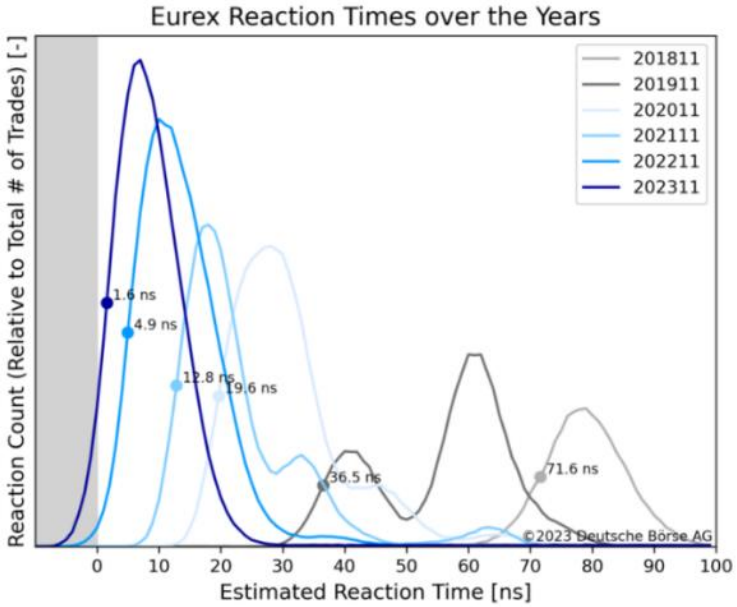
Corrupted Speculative Triggering (CST)

The minimum reaction time between the reception of a market data packet and the sending of an order in reaction has decreased, for the fastest participants, from 20 nanoseconds to 2 nanoseconds in the last 3 years as can be observed on the below graph (Deutsche Börse source).

Starting 2022, **negative reaction time** (grey zone of graph) even started to be observed, which is physically impossible (it would consist of going back in time) and would break the principle of causality: it is indeed impossible to react to a random event before it occurs, considering the qualitative and quantitative limits imposed to the packets one is allowed to send.

We therefore hypothesized that some market participants were sending packets continuously, regardless of the receipt of market data, in order to have a packet or piece of packet ready to be used to build a conformant order when a market data arrives.

Following many test and trials in laboratory and in production, we evidenced in detail the practice of Corrupted Speculative Triggering that breaches not only EUREX's rules and regulations but the Ethernet standard.



The principle of the CST is to send continuously non-conformant, **shortened Ethernet packets** at a rate of tens of millions per second that will be used in the Speculative Triggering process to build orders **in advance of** the reception of the market data and in advance of any participant behaving orderly.

In the "regular" Speculative Triggering process, outgoing Packets are sent **only when** an incoming Market Data packet is received. Consequently the outgoing Packet can only start to be sent at time T0 of the reception of a market data, as described by EUREX in Document 1 page 10.

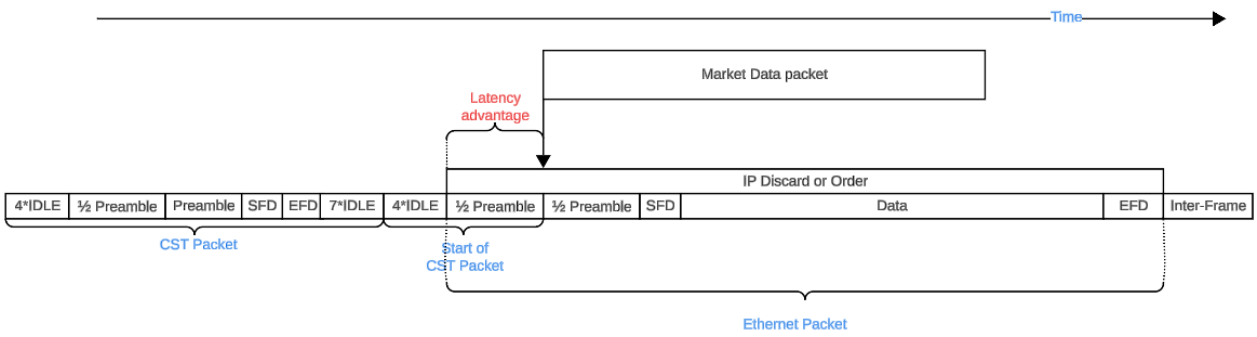
With the CST technique however, **shortened packets are sent in a continuous way**, irrespectively of whether market data is incoming.

The structure of CST packets combine a non-conformant sequence of Preamble (or part of a Preamble as we will see), SFD, EFD, and the data (Frame) length does not respect the 64 bytes minimum.

An example of CST packets we have tested, sent in continuous, can be represented in the following way:

When a market data packet actually incomes, the algorithms performing the CST technique can **use part of** the last CST packet sent (the subpart containing the ½ Preamble) to build an outgoing conformant packet.

Simplified representation of an Order being sent using the CST technique:



If a market data occurs, the algorithm will use the ½ Preamble of the last CST packet sent, add another ½ Preamble to make a full Preamble, and continue building the Order.

This ½ Preamble has a length of 4 bytes which is 4 x 8 = 32 bits. Sent on a 10Gbits optical line, it gives a 3.2 ns advantage. Orders built with this technique gain systematic priority over orders sent at time T0 by participants behaving orderly.

We have not tested all possible formats of CST packets, however a CST packet is by construction conceived in order to remain undetectable, at least to the way EUREX monitors its network traffic. If a CST packet was conformant to the Ethernet protocol (containing a minimum 64 Bytes Frame, among other requirements), it would be automatically detected and counted by any of the counters inside the monitoring equipment.

All CST packets are intentionally violating the rules defined in the NAG (*in italic)* in at least 3 ways:

- They are "*Shortened Ethernet Packets*", non-conformant to the Ethernet standard which requires a minimum size of 72 Bytes.
- They contain a "*corrupted Ethernet frame*" as the SFD (Start of Frame Delimiter) declares a Frame which should compulsorily be composed of a minimum of 64 bytes of data.
- These packets are "*not reaching the gateway*"

Why are CST packets not detected by EUREX ?

We have learned that EUREX argues that CST is in compliance with its Network Access Guide as long as it does not trigger an alarm in its monitoring procedures.

EUREX apparently does not enforce the NAG with regard to the qualitative requirement of the packets sent by participants and only relies on the '*Ingress Sequence number extension by Arista 7130 for monitoring purpose'*.

This information was added by EUREX only recently, on July 16th 2024, in version 2.3.3 of the NAG and subsequent versions, following our formal notification of the non-conformance of practice to EUREX in June 2024.

CST packets being intentionally built so as to remain undetected (exploiting loopholes in the Metawatch firmware of the ARISTA 7130 which should detect and count any packet not respecting the Ethernet protocol) they indeed do not raise any alarm in the Metawatch software, at least in its standard configuration.

CST packets are also intentionally built so as to systematically be discarded by the Cisco Switch (whose role is to forward the orders up to the matching Engine).

In order to enforce the rules set in the NAG, there are several external tools available that could be implemented by EUREX (we even developed one in FPGA to be used in real time) in order to perform in depth and real-time analysis of network traffic.

It is however not necessary: we evidence further in this document that in the existing Arista equipment, there already exist specific counters, which reveal and count the CST packets sent in millions.
We informed EUREX about the existence of these counters in a technical document we sent them. EUREX did not reply.

Recognition of the CST practice by EUREX

In the course of a law suit against the German regulator ESA (Exchange Supervisory Authority of the state of Hessen) at the administrative Court of Wiesbaden, whose conclusions have been made public, EUREX has admitted that (free translation from German):

"*A more advanced form of speculative triggering is known as Advanced Speculative Triggering (AST). Here, two data records, the so-called preamble and the "Start of Frame Delimiter" (SFD), are sent continuously - i.e. before a market data message is received. The preamble and the SFD are not part of the Ethernet frame mentioned above. Rather, they are prepended to the Ethernet frame in a complete data packet. This is already evident from the term "Start of Frame Delimiter" (German: Begrenzungszeichen für den Beginn des Frames).*
When a market signal arrives, the preamble and the SFD can then be directly linked to an Ethernet frame and converted into a reaction order from the trading participant. Compared to speculative triggering, in which the market signal is at least partially awaited, AST thus enables further latency optimisation; the latency here is in the low single-digit nanosecond range."

(Note: AST is the original acronym MOSAIC gave to the CST practice in our discussions with EUREX.)

EUREX has thus recognized the existence of the CST practice and has even provided a particular format of CST consisting in [Preamble + SFD], sent continuously for the purpose of gaining latency.

Considering that a conformant Ethernet packet must follow the structure [P + SFD + data + EFD] and must have a minimum length of 72 bytes, we must first note that the CST packets, as described by EUREX, violate at least § 12.1.1 of the Network Access Guide which prohibits the sending of "**Shortened Packets**".

This obvious (among others) violation of EUREX rules should be enough to ban the practice.

More interesting is that, in addition to this violation of the NAG, we have tested this sequence [P + SFD] described by EUREX in our laboratory consisting of a replication of EUREX network and using the same ARISTA and CISCO equipment.

Our tests revealed that EUREX's technical description of the CST cannot be the actual format performed in production (see Annex III)

CST packets are in reality a much more "creative" sequence of data in order to be at the same time:

- Undetected by the counters EUREX is currently monitoring
- Discarded by the CISCO switch when sent "alone"
- Able to cross the CISCO switch when a conformant Ethernet packet is built using part of the CST.

EUREX has consequently transmitted misleading information to the administrative Court of Wiesbaden and the question that arises is whether (and which) market participant(s) transmitted this false information to EUREX.

Possible and most probable format of the actual CST format performed on EUREX

We have made extensive tests and identified several ways to construct a CST packet.

The most probable format actually performed, considering its ability and stability to reach the goals of being both undetectable by the Metawatch firmware of ARISTA 7130 (not counted when sent "alone") and able to however cross the CISCO switch when a conformant packet is built using part of the CST, has the following structure:

4I +1/2P + [P+SFD] +EFD + 7I

Where:

- "I" is Idle
- "P" is Preamble
- "SFD" is Start of Frame Delimiter
- "EFD" is End of Frame Delimiter

This format of CST is undetected by the counters that EUREX is currently monitoring and can be used to build orders with a 3.2 nanoseconds in advance to the reception of a market data packet.

We have submitted this CST packet [4I + P + SFD + EFD+ 7I] to the same test bench described in Annex III and confirm that:

- These packets, when sent alone, are not counted by counters EUREX is currently monitoring (Ingress sequence number)
- These packets are discarded by the CISCO equipment when sent alone.
- A conformant Ethernet packet can be completed "on the fly" **using the [1/2P] sub-part of this CST**, which will cross the CISCO switch and reach the matching engine of EUREX if necessary.

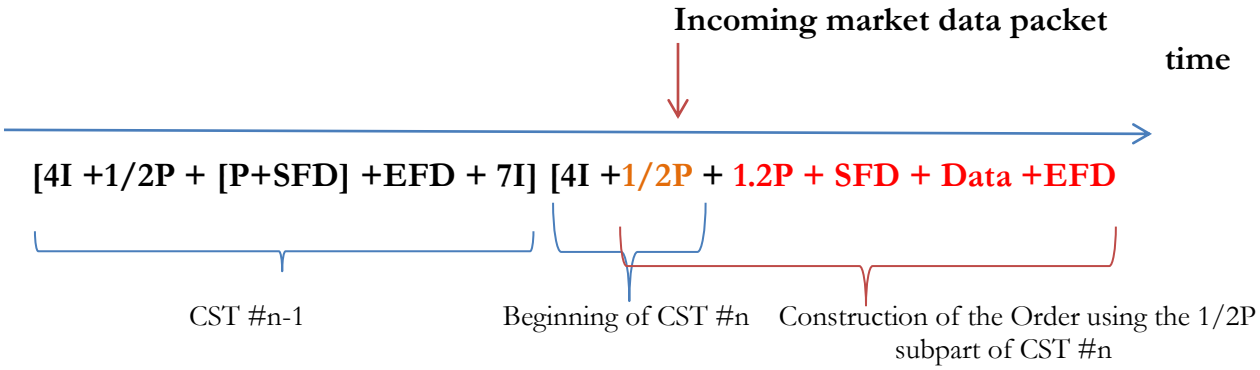
How it works in practice:

The size of the CST packet we describe is 24 Bytes:

(4I = 4 bytes, ½ P = 4 bytes, P = 7 bytes, SFD = 1 byte, EFD= 1 byte, 7I = 7 bytes).

This packet is sent continuously on the 10Gbit/s optic fiber Order Entry connection.
Considering there are 8 bits in a Byte, this CST Packet (which has a length of 24 x 8 = 192 bits) is sent at the frequency of **52 083 333 per second** ($=10^{10}/192$).

When a Market Data Packet incomes, the algorithm performing this format of CST will add a [1/2P +SFD] to the existing [1/2P] of the last CST Packet being sent, and then complete the Order (or IP discard) with the data required for a conformant Ethernet Packets.



The latency advantage of the Order in construction is given by the subpart [1/2P] of the CST that has already been sent before the Market Data incomes.

The size of [½ P] being 4 bytes (= 32 bits), on a 10 Gbit/s connection the latency advantage of the Orders sent with the CST technique is $32/10^{10} = 3.2$ nanoseconds.

The violation of this CST format to EUREX rules (NAG) are the same as those of the wrong CST format [P + SFD] given by EUREX:

- It is a "*Shortened Ethernet Packets*", non-conformant to the Ethernet standard which requires a minimum size of 72 Bytes.
- These packets are "*not reaching the gateway*"
- It contains a "*corrupted Ethernet frame*" as the SFD declares a Frame which should compulsorily be composed of a minimum of 64 bytes of data.
- It is sent at a frequency (here 52 083 333/s) exceeding the maximum of 30 000/s.

In order to confirm our analysis of the non-conformity of this CST format, we required an analysis from a third party, internationally recognized expert, the InterOperabilityLaboratory of the University of New Hampshire (IOL).

We sent the IOL 3 different formats of possible CST implementation, including the format described in this chapter.

The IOL concluded that all 3 formats violate the IEEE 802.3 standard, also known as Ethernet protocol (see Annex I).

How to detect CST within the existing monitoring equipment used by EUREX.

CST packets have been designed in order to exploit loopholes of EUREX monitoring procedures and remain undetected.

With respect to the current EUREX monitoring approach (relying on the Ingress sequence number of the METAWATCH software), not only are these corrupted packets not detected as corrupted (qualitative breach

to the NAG), but they do not increment the Ingress Sequence number monitored by EUREX to determine the number of frames sent per second (quantitative breach to the NAG).

However, inside the Arista equipment, we found out that several counters are available, some of them are provided by the METAWATCH software (the Arista development team is mastering them) and monitored by EUREX, some others are provided (according to our technical understanding) by the physical interface component MARVEL WPCS4317C.

The WPCS4317C from MARVEL is a 10 Gbit/s physical interface capable of deep line inspection such as, among others, compliance with the IEEE 802.3 standard. It integrates nearly 640 counters per line for monitoring purposes, including specific counters dedicated to IEEE 802.3 Clause 49, which can reveal CST behavior.

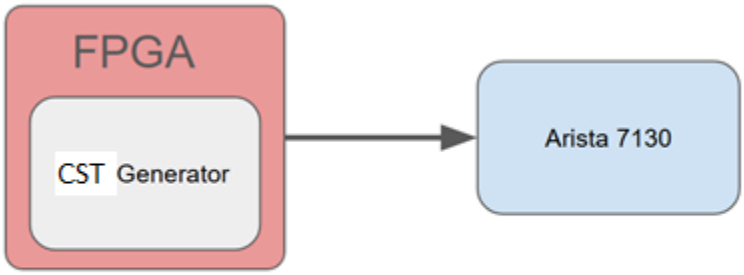
In this document we evidence a method to detect the CST packets, by looking at counters from MARVEL components that are accessible, just like the FPGA ones, that is to say via simple CLI (Command Line Interface) command on the Arista equipment itself.

We informed EUREX in April 2025 of the existence of these counters. Despite our repeated offers, they refused to comment.

The test bench setup

The detection process we propose is effortless and does not require a modification of EUREX's monitoring infrastructure (such as a new derivation): the Arista equipment remains as the sole monitoring equipment.

An FPGA capable of generating various types of CST packets has been connected to an ingress port of the Arista equipment, and by using the available monitoring functionalities of the Arista, we found a way to detect and count the CST packets and frames.



CLI command to be performed to evidence the CST packets and count the Frames.

On the terminal of the ARISTA equipment, by typing this CLI command:

show int et[interface number] counters verbose

A set of counters are shown, and we can focus on "line_pcs49_rx_good_frames" and "line_pcs49_rx_err_frames":

- **line_pcs49_rx_err_frames**: counts the number of **"Start"** control character without **"Terminate"** control characters
- **line_pcs49_rx_good_frames**: counts the number of **"Start"** control character followed by **"Terminate"** control characters

These counters explicitly count frames, which is consistent with the IEEE802.3 standard that describes as mandatory that an Ethernet packet embeds an Ethernet frame.

The test results

We performed extensive tests on various types of CST packets and well-formed packets. We split results in 2 major cases:

- Case 1: The standard case, sending of well-formed Ethernet packets
- Case 2: The continuous sending of the CST packets with the format described in this paper

Case1: sending of well-formed Ethernet packets:
We sent 106 standard (well formed) Ethernet packets in an orderly manner, with standard "Idle" control characters in between packets. We observe the following results for the counters:

line_pcs49_rx_good_frames: 106
line_pcs49_rx_err_frames: 0

```
lab-mux03(config-app-metawatch)#clear counters
lab-mux03(config-app-metawatch)#show int et2 counters verbose | grep line_pcs49_rx_good_frames
Collecting all statistics for port et2
      "line_pcs49_rx_good_frames" : 106,
lab-mux03(config-app-metawatch)#show int et2 counters verbose | grep line_pcs49_rx_err_frames
Collecting all statistics for port et2
      "line_pcs49_rx_err_frames" : 0,
```

Case2: continuous sending of the CST packets with the format described in this paper:
This CST packet has a size of 24 bytes = 192 bits that gives a period of 19,2 ns. The frequency of the AST packets is 1 second / 19.2 ns = 52 083 333 packets/s
We cleared the counters "**clear counters**" and then captured the counters after ~10 seconds. We observe:

line_pcs49_rx_good_frames: 0
line_pcs49_rx_err_frames: 512 314 840
This correspond to the frequency at which CST packets were sent (52 083 333 packets/s)

```
lab-mux03(config-app-metawatch)#clear counters
lab-mux03(config-app-metawatch)#show int et2 counters verbose | grep line_pcs49_rx_err_frames
Collecting all statistics for port et2
      "line_pcs49_rx_err_frames" : 512314840,
```

We have also tested various versions of CST packets, some of which are counted as "good frames" and some as "error frames".

Regardless the format of the CST packet, as it necessarily begins with a **"Start"** (meaning Start of Ethernet packet) control character, frames will be counted by the MARVEL counter (either as a "good" or an "error" frame").

In any case ("good" or "error" frame), on a particular connection, the sum of "line_pcs49_rx_err_frames" + "line_pcs49_rx_good_frames" **reveals the actual number of frames**.

In order to further confirm these counters are not recent features and not dependent from a firmware or hardware version of the Arista, we have conducted these tests using various versions of the Metawatch firmware we have available on the Arista 7130, as well as on older products of the same vendor: the Metamux 32 and Metaconnect 48 equipment.

These counters have always been present and gave us the exact same results as those presented above.

This is explained by the fact that these counters come from the MARVEL WPCS4317C component, consistently present on the Metamako / Arista equipment we have tested.

The EUREX's monitoring equipment (Arista 7130) does have built-in capabilities of detecting CST packets. No configuration change is necessary.

The only thing that has to be done is reading the already existing and functional counters:

line_pcs49_rx_good_frames
line_pcs49_rx_err_frames

This further confirms that CST packets increment Frames counters even within the EUREX current monitoring equipment.

We additionally propose in Annex II a real monitoring set up to be easily implemented.

How CST increases EUREX Order Entry connectivity revenues by 3.

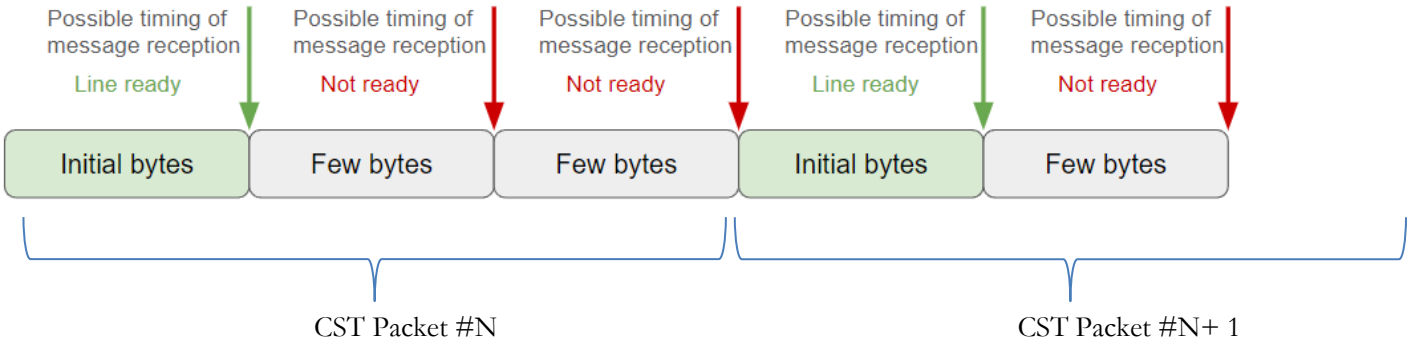
Any CST packet giving a 3.2 ns gain is constructed on the following pattern: [Initial Bytes] + [Filler bytes]

- The "Initial bytes" is ½ Preamble in the CST example analyzed in this document. It can have other shapes. These "initial bytes" are the bytes that systems using CST are trying to "pre-send" so that as soon as a message is received from the market, some bytes have already been sent, hence gaining a time stamping in advance of market participants behaving orderly.
  It is to some of these "initial bytes" that a complete frame is aggregated at the reception of a market message, in order to build a market order (or an IP discard).

- The "Filler bytes" can also take various shapes. We thus provide a generic description here. They could be for example sequences of EFD (End of Frame Delimiter) or Idle. They could contain small quantities of data (< to the required 64 bytes). They could even be repetitions of Preamble. The goal of these filler bytes is solely to "confuse" the switches in such a way that the switches discard these CST packets seamlessly, while keeping the possibility to have packets properly formed when attaching a proper frame to the "initial bytes" pre-sent.

We have identified several functional CST patterns and they all follow such a sequence. No CST pattern has been identified where simply sending "initial bytes" continuously is functional (as EUREX would like to make believe the Administrative Court in its description of the CST), hence the need for these filler "few bytes".

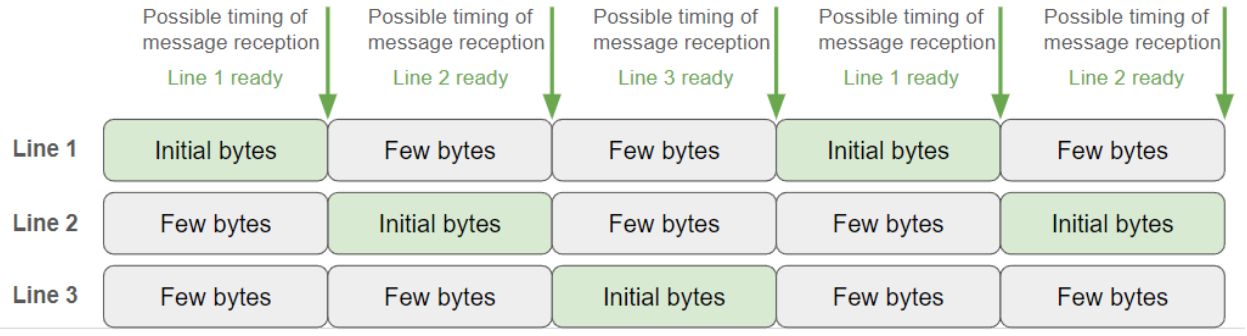
The following schematic displays the typical sequence of bytes continuously sent constituting CST packets:



Market data packets (represented above with vertical arrows) can occur at any time.

There is no way to predict at which moment they may be received. Moreover, it is only when they occur at the exact timing when an "initial bytes" has been sent, that there is the possibility to aggregate data to the pre-sent "initial bytes" in order to perform a regular packet . As a result, with one line, considering a pattern of 3 repeating blocks for a CST pattern (which is the minimum size pattern found so far and the format of the example of our document consisting of 3 blocks of 8 bytes), an "initial bytes" is available only one third of the time.

This is not ideal but the workaround is simple:



By having 3 different Order Entry lines and shifting the CST pattern by 1 block for each line (asynchronous **sending of 3 CST packets on 3 different Order Entry lines), the system is now ready to respond to market data packet at any time**, as there is always one of the lines that just sent an "Initial Bytes".

The participant performing CST must consequently subscribe and pay to EUREX 3 times the quantity of Order Entry connections as it would be necessary without CST.

There are currently 864 Order Entry lines proposed to market participants in collocation.
One Order Entry connection is typically billed around 72k€ per year by EUREX, which is over 62 M€ potential revenues per year.

We can see below that since March 2024, only new Order Entry lines have been added by EUREX (no additional Market Data line).

In our opinion this increased number of Order Entry connection is mainly dedicated at responding to the demand of participants performing CST on a large scale.

Number of **switches**	Room A		Room B		
In production at date:	Market Data	Order Entry	Market Data	Order Entry	Total
End of 2022	4	5	6	6	21
13/03/2023	5	6	7	6	24
23/03/2024	6	7	7	7	27
22/02/2025	6	9	7	9	31

Number of **lines**	Room A		Room B		
In production at date:	Market Data	Order Entry	Market Data	Order Entry	Total
End of 2022	192	240	288	288	1008
13/03/2023	240	288	336	288	1152
23/03/2024	288	336	336	336	1296
22/02/2025	288	432	336	432	1488

Conclusion

In this document we revealed that EUREX has admitted the existence of the practice consisting in sending continuously shortened packets in order to make latency gains on market orders built with this technique.

The fact that CST is performed on EUREX is consequently not a questionable issue.

We however have evidenced that the technical format of these packets, as given by EUREX, is not the actual format performed. This questions the origin of this information given by EUREX.

We have provided readers with a format supposedly performed by some ULL participants and confirm that this format:

- is undetectable in the current EUREX's monitoring procedure,
- is discarded by the CISCO switch when sent alone,
- can be used to perform genuine market orders with an advance of 3.2 nanoseconds to the actual reception of market data.

Although we cannot guarantee that it is the CST format actually implemented (only EUREX can have access to this information) we strongly believe it is the most common form of CST performed.

Whatever the format of a CST packet, it is necessarily corrupted with regards to the Ethernet standard: if a CST format was a conformant packet with a regular size, it would be automatically detected by the current EUREX monitoring equipment and breach the frame limits.

Any CST packet format breaches most of NAG rules:

- It is a "*Shortened Ethernet Packets*", non-conformant to the Ethernet standard which requires a minimum size of 72 Bytes.
- These packets are "not reaching the gateway"
- It contains a "*corrupted Ethernet frame*" as the SFD declares a Frame which should compulsorily contain a minimum of 64 bytes of data.
- It is sent in millions (52 083 333 in our example) per second, breaching the maximum allowed of 30 000 frames/s.

We have evidenced that within the EUREX monitoring equipment Arista, there exist frame counters which are incremented by CST packets, and even qualified either as "good" or "error" frames. EUREX turns a blind eye on it.

We explained how EUREX financially benefits from the CST technique which maximizes its Order Entry connectivity business.

In light of these substantial financial benefits, EUREX's acceptance of a practice that so clearly contravenes its own policies raises the possibility of a conflict of interest.

Annex I: Extract of IOL report

Note: 3 different formats of CST have been analyzed by the IOL at MOSAIC's request. "Frame capture1.csv" is the CST format example described in this document (4I + 1/2P + P + SFD + EFD + 7I).

10GBASE-R PCS Analysis Report Test Result ID: 39144

SUMMARY OF FINDINGS

Included in this document is an analysis of 66B vectors, provided in a form whereby they have already been received, deserialized, descrambled and aligned to Sync Headers. Three captures were provided to determine if the sequences represent valid encodings of 64B66B per the IEEE Standard 802.3-2022.

From the provided 66B vectors, it is apparent that the sequence "Frame capture1.csv" represent illegal sequences, per the requirements of the IEEE 802.3-2022 subclause 49.2.4 *64B/66B transmission code* requirements. Specifically, the PCS Transmit State Machine is violated, as any device that was commanded (at the XGMII level) to send the sequence shown below should (if conformant) have sent an "EBLOCK_T" as the Transmit State Machine would require the device to do so when in the TX_E state.

For example, the pattern seen in "Frame capture1.csv" could not be sent by a conformant device, as such a device should send an EBLOCK_T as shown below, rather than the observed 66B vector (1078555555555555D5).

Raw 66B block from capture	Block Type	TX State	TX State action	**Required** 66B block
101E00000000000000	[C | C | C | C][C | C | C | C]	TX_C	tx_coded <= ENCODE(tx_raw)	As provided by XGMII
103300000000555555	[C | C | C | C][S | D | D | D]	TX_D	tx_coded <= ENCODE(tx_raw)	As provided by XGMII
1078555555555555D5	[S | D | D | D][D | D | D | D]	TX_E	tx_coded <= EBLOCK_T	101E1E1E1E1E1E1E1E
108700000000000000	[T | C | C | C][C | C | C | C]	TX_T	tx_coded <= ENCODE(tx_raw)	As provided by XGMII
103300000000555555	[C | C | C | C][S | D | D | D]	TX_D	tx_coded <= ENCODE(tx_raw)	As provided by XGMII

Refer to the "File Analysis" section for further discussion and background information. Refer to the Appendices for full capture information and a mapping to likely results if these observations were provided in a UNH-IOL 10GBASE-R PCS Test Report for 64B66B encoding.

While the inarguable standard's violations are summarized above, the inter-packet gap (IPG) sent by a device should be idle, per IEEE 802.3-2022 subclause 46.2.1 "the MAC interpacket gap begins with the Terminate control character, continues with Idle control characters and ends with the Idle control character prior to a Start control character." In all three provided captures, the IPG is atypical, as it should be "Idle" (encoded as 101E00000000000000 , which is a full control block type ([C | C | C | C][C | C | C | C])

It is possible, but speculation, that the provided captures are from a solution attempting to lower latency by 'pre-sending' the first 4 to 8 bytes of a frame (which, at the Ethernet MAC level must always commence with 7 bytes of preamble and 1 byte Start of Frame Delimiter (SFD)) Such a solution may certainly trigger interoperability issues due to the standard's violations noted above. As an Ethernet MAC should always generate the required preamble and SFD, such a 'pre-sending' solution seems of limited benefit unless the Ethernet MAC is also non-conformant.

The IPG limitations imposed by the Clause 49 64B66B PCS layer (notably the restriction that a frame can only start on a 4-byte boundary) can be eased should an implementation wish to implement the Deficit Idle Count (DIC) methodology, refer to subclause 46.3.1.4(2). Should an implementation wish to reduce latency, implementing this standard's compliant IPG shrinkage mechanism is a path with a greater chance of interoperability.

Please refer to the Conclusions for additional discussion on two other violations of note, the illegal inter-packet gap present in all three captures analyzed, and the corresponding illegally small Ethernet Packets sent. Both of which are sent in significant numbers immediately before the well-formed packet is transmitted, leading to likely interoperability issues with compliant link partners.

CONCLUSIONS

The first capture provides strong evidence that the source of the 66B sequence, if properly captured and decoded in the provided capture file, is violating the IEEE 802.3-2022 Clause 49 PCS Transmit State Machine requirements.

All three captures demonstrate irregular inter-packet gap (IPG) behavior. While the Summary of Findings provides some speculation on the possible motivations for such behavior, this conclusion will simply remind the expectations on IPG per the IEEE Standard:

> As outlined in subclause 46.2.1 *Inter-frame <inter-frame>*
>
> " *The <inter-frame> corresponding to the MAC interpacket gap begins with the Terminate control character, continues with Idle control characters and ends with the Idle control character prior to a Start control character.*"

There are many requirements on interpacket gap timing, and while under certain circumstances, the inter-packet gap (IPG) may shrink to as little as 5 bytes (see 46.2.1) at the RS layer, the behavior on the line should be such that the minimum IPG is no less than 12 bytes when transmitting frames. The one exception to this is the allowance for the use of Deficit Idle Count (DIC), specifically created to allow an average IPG of 12-bytes for solutions, such as the Clause 49 PCS, that restrict frame transmission to 4-byte boundaries (refer to subclause 46.3.1.4(2)), whose implementations without DIC would average above the minimum IPG in most scenarios.

In addition to the evident PCS Transmit State Machine requirement violations observed, it is further speculated that any implementation attempting to reduce latencies in the manner observed may also be internally violating the Ethernet MAC requirement to generate 7 bytes of preamble and 1 byte of SFD for each frame that it wishes to send (see subclause 4.2.8 *procedure* PhysicalSignalEncap for normative MAC requirement). It should be noted that the only enforceable standard's compliance requirement is that which can be observed externally (on the Ethernet medium in this instance), hence any speculation on internal device behaviors is not overly productive. All 3 frames observed (one in each capture) are encoded with valid preamble and SFD (1 byte /S/, 6 bytes of preamble (0x55) and one byte SFD (0xD5) as observed on the medium) and are terminated properly per 64B66B.

At issue is the coding prior to the Start character (/S/) of each of these frames, which was preceded by 6,528 Bit Times (BT) {capture 1}; 6,784 BT {capture 2}; and 6,528 BT {capture 3} of non-XGMII idle signaling prior to the frame being sent. As these preceding sequences all contain some combination of /S/ and /T/ characters, these rapid indications of packet start and terminations may create interoperability issues, and themselves are violations of the minimum IPG requirements and minimum Ethernet Packet size requirements.

Even if not a clear violation of Clause 49 requirements, such signaling, received by a conformant Clause 49 PCS Receiver, will indicate to the Reconciliation Sublayer ("RS", above the XGMII, and immediately below the Ethernet MAC) that a frame is being received, and then prematurely terminating, all with illegally small inter-packet gaps. Such combined violations may certainly cause an "RS" layer to detect a local fault (see 46.3.4 quoting here:"*Upon recognition of a fault condition a PHY sublayer indicates Local Fault status on the data path. When this Local Fault status reaches an RS, the RS stops sending MAC data*"), and/or the MAC may not be able to accept packet indications illegally close together (violating the minimum IPG). Such illegally small packet indications should be occurring in any conformant

implementation receiving such signaling prior to the complete well-formed 74-byte Ethernet Frames transmitted in the captures provided.

Take for instance the sequence from the first capture:

[S | D | D | D] [D | D | D | D]
[T | C | C | C] [C | C | C | C]
[C | C | C | C] [S | D | D | D]
[S | D | D | D] [D | D | D | D]
[T | C | C | C] [C | C | C | C]
[C | C | C | C] [S | D | D | D]

This 3 66B block repeating sequence (which repeated a total of 34 times) would cause the RS to receive 12 XGMII (or equivalent) 32-bit words, shown in the table below, where the 1st indicates a frame has commenced, so the RS would send to the MAC a PLS_DATA.indication and a PLS_DATA_VALID.indication, this would continue for the next 32-bit word, then end on the 3rd 32-bit word, indicating the data and valid indications ceased on lane 0, but only 11 bytes later, in the 6th 32-bit word a /S/ is indicated again, then then in the 7th another /S/ is indicated (this results in what should be a TX_E state transition) the sequence then continues to repeat.

	XGMII Lane 0	XGMII Lane 1	XGMII Lane 2	XGMII Lane 3
[S | D | D | D]	/S/	/D/	/D/	/D/
[D | D | D | D]	/D/	/D/	/D/	/D/
[T | C | C | C]	/T/	/Idle/ (1)	/Idle/ (2)	/Idle/ (3)
[C | C | C | C]	/Idle/ (4)	/Idle/ (5)	/Idle/ (6)	/Idle/ (7)
[C | C | C | C]	/Idle/ (8)	/Idle/ (9)	/Idle/ (10)	/Idle/ (11)
[S | D | D | D]	/S/	/D/	/D/	/D/
[S | D | D | D]	/S/	/D/	/D/	/D/
[D | D | D | D]	/D/	/D/	/D/	/D/
[T | C | C | C]	/T/	/Idle/ (1)	/Idle/ (2)	/Idle/ (3)
[C | C | C | C]	/Idle/ (4)	/Idle/ (5)	/Idle/ (6)	/Idle/ (7)
[C | C | C | C]	/Idle/ (8)	/Idle/ (9)	/Idle/ (10)	/Idle/ (11)
[S | D | D | D]	/S/	/D/	/D/	/D/

Note this continued violation (as the number of Idle bytes is only 11, not 12) of the minimum IPG is, itself, a violation of transmit requirements (refer again to 4.2.3.2.2) however the impact on receive behavior and interoperability is more unclear as devices should accept frames below the minimum IPG (see again 46.2.1) however such a prolonged period of violations may trigger PHYs to indicate a local fault (see again 46.3.4) which is generally undefined by the standard (one obvious and expected 'local fault' condition would be a loss of signal).

Nonetheless, the illegally small packets are a final issue to note, as the smallest permitted Ethernet Frame size is 64 bytes, with the smallest allowed size from /S/ to /T/ being 72 bytes (adding preamble and SFD Packet size requirements to the minimum Frame size). While illegally small packets are less likely to cause issues with practical Ethernet systems, when combined with the illegally small IPG noted above, and the fact that many of these events immediately precede a frame that is, presumably, intended to be received by conformant Ethernet systems, the overall combination of violations, unsurprisingly, is likely to yield interoperability issues in conformant link partners receiving such sequences.

Annex II: Monitoring proposal

The Arista Metamako device includes a documented feature that enables periodic transmission of these counters to an external time-series database:

`destination influxdb` **(config-mgmt-telemetry)**

send metrics to an external influxdb instance via http or udp

```
destination influxdb http|https|udp ADDRESS|HOSTNAME PORT DATABASE
RETENTION_POLICY|default [USERNAME] [PASSWORD]
```

ADDRESS

IP the external influxdb instance

PORT

Port the externel influxdb instance is listening on

DATABASE

Database to post metrics to

Thanks to this, it is possible to build an effective monitoring system to detect the use of CST: all we need is an InfluxDB database to store the counters, and analytic software like Grafana to visualize the counters as graphs, define thresholds, and configure alerts.
To show how easily this can be done, let's build it. We are using Docker Compose to define the two components of our monitoring system within a single configuration file:

```
version: '3.7'

services:
  influxdb:
    image: influxdb:1.8
    container_name: influxdb
    ports:
      - "8086:8086"
    volumes:
      - influxdb-data:/var/lib/influxdb
    restart: unless-stopped

  grafana:
    image: grafana/grafana
    container_name: grafana
    ports:
      - "3000:3000"
    volumes:
      - grafana-storage:/var/lib/grafana
    restart: unless-stopped

volumes:
  influxdb-data:
  grafana-storage:
```

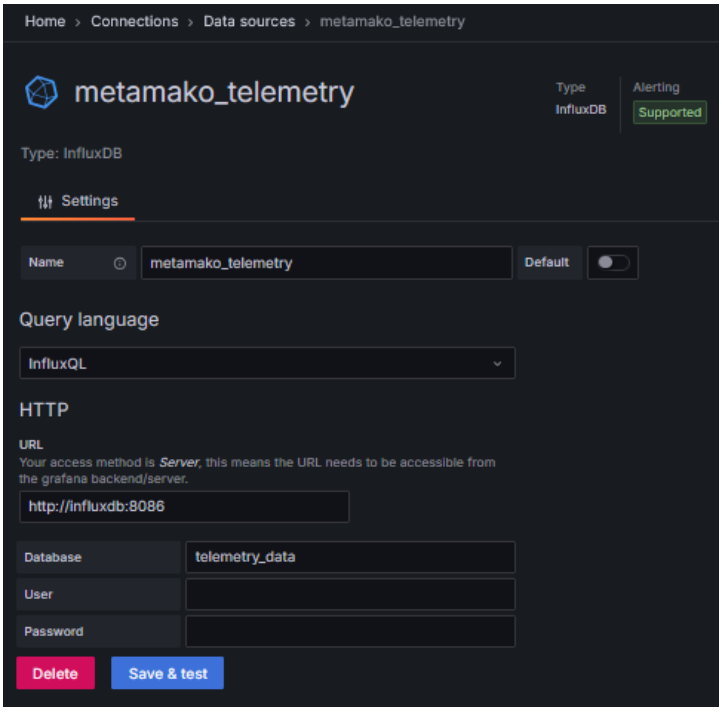
Let's run this and then create a database 'telemetry_data' in our new InfluxDB instance:

```
oot@lab01:/srv/ast-monitoring# docker compose up -d
root@lab01:/srv/ast-monitoring# docker exec -it influxdb influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> CREATE DATABASE telemetry_data
> exit
```

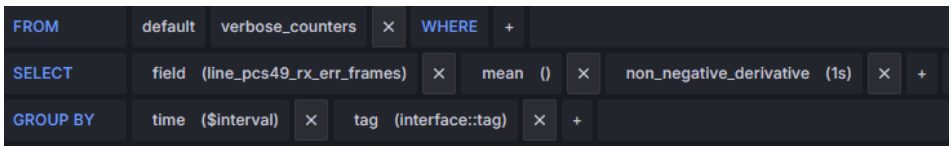
We can now apply the feature on the Arista Metamako to start sending the counters to this database:

```
mux-601A(config)#management telemetry
mux-601A(config-mgmt-telemetry)#destination influxdb http 10.X.X.X 8086 telemetry_data default
```

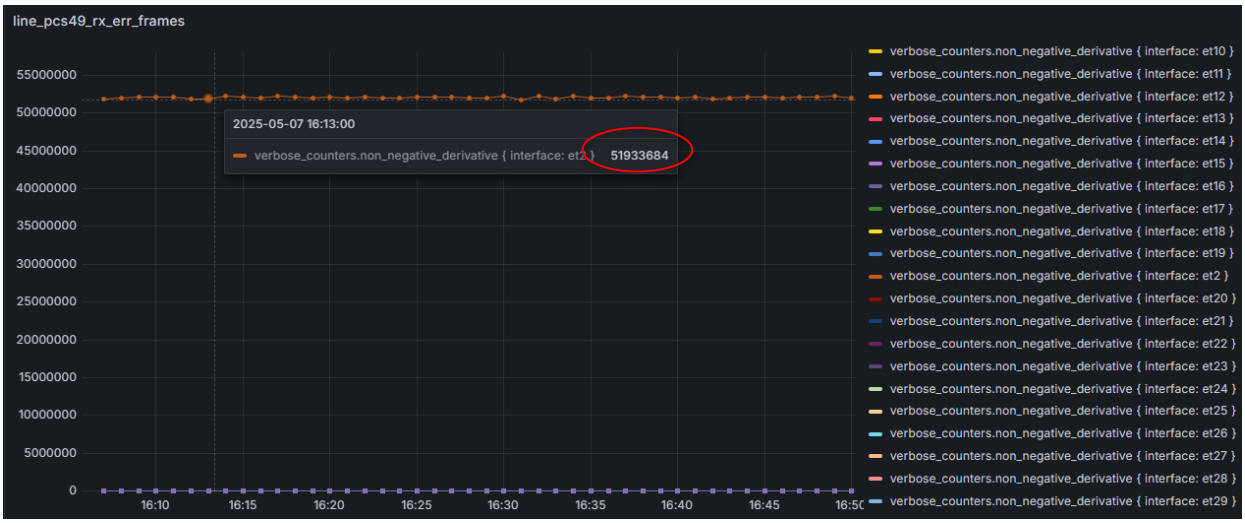
The data is now being collected, and we just need to set our database as an InfluxDB data source in our analytic software Grafana:



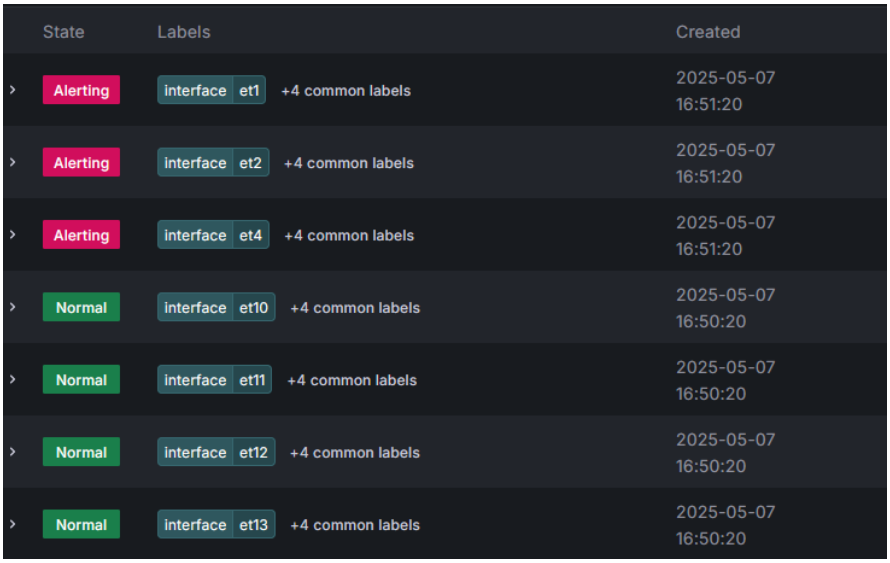
It is now possible to create graphs in the dashboard section showing the line_pcs49_rx_good_frames and line_pcs49_rx_err_frames for all interfaces on our Artista Metamako:



The result is a *graph clearly showing one interface (et2) with an average of 52 000 000* errors per seconds matching the frequency of the CST pattern tested:



With a couple more click in the same software, it is possible to set-up automatic alerting if a counter goes over a defined threshold.



Example of the alerting section of Grafana, showing the 3 interfaces on which we sent CST packets (et1, et2 and et4) breaching the defined threshold.

Annex III: Testing of the CST format [P + SFD] as declared by EUREX.

We programmed a CST generator in FPGA, sending continuously the CST Packet described by EUREX [Preamble + SFD], and every 100 milliseconds data is added right after one of these Packets in order to replicate the building of a properly built, complete (containing at least 64 bytes of data) Ethernet packet (IP Discard or Order).

At PCS layer (Physical Coding Sublayer) the only way to represent a continuous transmitting of [Preamble + SFD] is a set of data of 8 Bytes coded the following way:

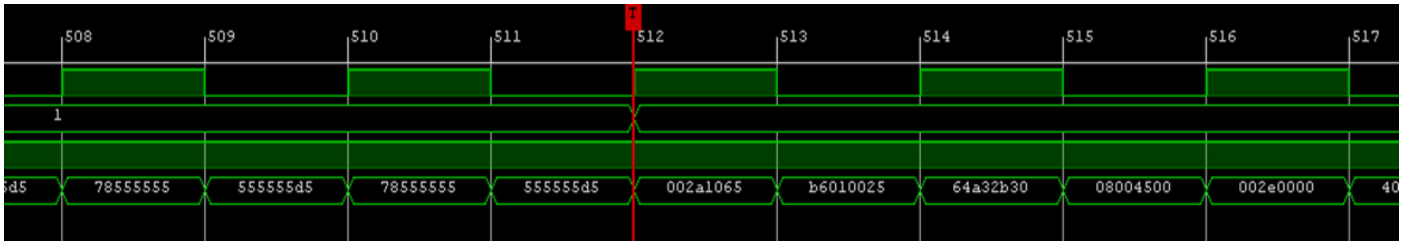
x"78_55_55_55_55_55_55_D5"

(The X"78" control code is the "start" control character and the start of the Ethernet Packet).

Consequently our test FPGA has been set up in the following way:

- Burst of 100 continuous CST Packets x"78_55_55_55_55_55_55_D5"
- After a random number of CST packets sent alone, adding of Data (size > 64 Bytes) following a CST Packet, in order to form a proper, complete Ethernet Packet

The sequence above is repeated continuously.



The results of the test are the following:

- We observe that the ARISTA equipment behaves as expected with regard to the goals of the CST technique: Ingress counters are properly incremented by 1 each time a properly formed packet is sent, and the CST packets sent "alone" are not counted.
- However the CISCO equipment, when receiving this type of CST packets discards every packet it receives, **even the well-formed packets don't get to exit the CISCO** and will never reach the matching engine.

Consequently the [P + SFD] format of the CST provided by EUREX cannot be the one actually implemented by participants performing CST.

Annex IV: Why it is impossible to pre-send orders respecting the Ethernet standard and the limitation of 30 000 frames / second, in order to gain latency.

We observe a maximum 10 000 market data updates (incoming messages) per second on index products, as an example (*see note, bottom of this page).
In average, between 2 market updates there are consequently 1/10 000 s = 100 000 ns

A well-formed Ethernet packet has a minimum size of 72 bytes = 576 bits, that is a 57.6 ns period.
In 100 000ns it is consequently possible to send a maximum of 100 000 / 57.6 = 1736 regular packets

Being allowed to send only 30 000 packets per second, that is 3 per 100 000ns, a participant performing "regular" speculative triggering and trying to send randomly a quota of 30 000 Ethernet packets/s in order to match the arrival of the market update, would only have a 3 / 1736 = 0.176 % chance to succeed.

* See EUREX analysis page 35 of the following document which indicate that for example the maximum cumulated volume of incoming frames (market data) on the main equity index product (FESX, FDASX, OESX, ODAX) is 400 000/minute, which is 6666/second, that we round to 10 000/second to minor our mathematical proof.
https://www.eurex.com/resource/blob/48918/9d4d29a403418f093c584b48c43990a7/data/presentation_insights-into-trading-system-dynamics_en.pdf