Reliability in deterministic networks: Comparison of FRER (TSN) and PREOF (DetNet)

Marta Blanco Caamaño*, Luis M. Contreras*, Javier Velázquez Martínez*, Rafael A. López da Silva*
** Telefónica Innovación Digital, Madrid, Spain*

ABSTRACT
6G systems will need to make faster and more reliable decisions at the network edge to surpass the specifications outlined for eXtreme URLLC (xURLLC) services. One fundamental requirement concerning connectivity is determinism. Currently, a solution for incorporating determinism in Layer 2 is enabled by TSN, while in Layer 3 it is targeted by DetNet. Moreover, one possible option for forwarding in the case of DetNet is precisely the utilization of TSN as a data plane [1]. Hence, there could potentially exist a DetNet/TSN domain in real deployments. Both standards present techniques to enhance network reliability through packet replication and elimination, as FRER [2] in the case of TSN and PREOF [3] in the case of DetNet. This paper provides a comparison of both techniques.

Keywords: 6G, URLLC, TSN, DetNet, deterministic communication.

1. INTRODUCTION

Currently, it is possible to find solutions integrating determinism into Layer 2, is the IEEE Time-Sensitive Network (TSN), and into Layer 3, it is the IETF Deterministic Networking (DetNet). DetNet emerges as an extension of TSN at Layer 3 level, offering the ability to work with TSN in the forwarding layer, among other options such as MPLS-TE. Therefore, it would be possible to have both TSN and DetNet within the same network, simultaneously operating. Within the array of techniques presented by both TSN and DetNet to achieve deterministic behavior across the network, there are specific techniques aimed at enhancing one of the fundamental aspects for achieving determinism, which is network reliability. Both techniques, independently, aim to reduce packet loss in the network through packet (for DetNet) or frame (for TSN) replication, in the delivery, and elimination, at the reception, of the data.

In the case of TSN, this technique is referred to as Frame Replication and Elimination (FRER). It implies decreasing frame loss rates for a flow by assigning sequence numbers and replicating each packet at the source end system and/or relay systems within the network, delivering them through separated disjoint links, and subsequently eliminating these duplicates at the destination end system and/or other relay systems.

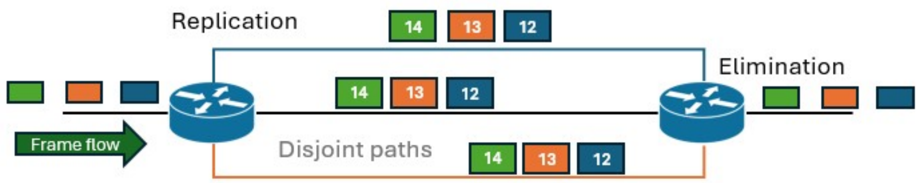

Figure 1. Example of FRER.

FRER natively provides five functions, but not all are required in every node.

- Sequencing function. Each data frame involved in FRER will have a sequence number, which will serve to detect latency errors in the different streams and to discard duplicate frames.
- Stream splitting function. This involves replicating frames and distributing them across various paths towards the same destination.
- Individual recovery function. It is the function responsible for discarding frames based on the sequence number.
- Sequence encode/decode function. It is responsible for the encoding, extraction, and removal of the sequence number carried by the frames.
- Stream identification. It is necessary to identify the streams to distinguish different traffic flows.

On the other hand, in DetNet, Packet Replication, Elimination, and Ordering Functions (PREOF) emerge to enhance network reliability. PREOF encompasses four capabilities:

- Sequencing information is provided to the packets of a DetNet compound flow. This may be achieved through the addition of a sequence number or time stamp as part of DetNet. This is typically done once, at the source node.
- Replication of these packets into multiple different paths to the destination (Packet Replication Function [PRF]).
- Elimination of duplicate packets of a DetNet flow based on the sequencing information and a history of received packets (Packet Elimination Function [PEF]). The output of the PEF is always a single packet.
- Reordination of a DetNet flow's packets that are received out of order using the sequencing information (Packet Ordering Function [POF]) [4].

FRER and PREOF are essentially distinguished by three fundamental characteristics. The first is that FRER, being a feature of TSN, operates at layer 2, whereas PREOF operates at layer 3. The second aspect is that, when identifying the various frames or packets, FRER requires the addition of an extra header. This necessitates an augmentation in the packet processing time at the nodes, as the inclusion of this supplementary information entails extending the packet and appending bytes for the new header. Additionally, this may lead to the need to fragment the frame. In contrast, in DetNet, both packet and flow identification are included in two fields of the DetNet stack, as seen in Figure 2. Therefore, the processing time of a DetNet packet at a node is a bit less than that of a TSN frame. The last distinction between the two techniques is that PREOF offers the functionality of ordering, which is not present in FRER. This functionality is crucial because there could be a scenario in the network where a link fails and then recovers, which may result in unordered arrival of packets at the destination.

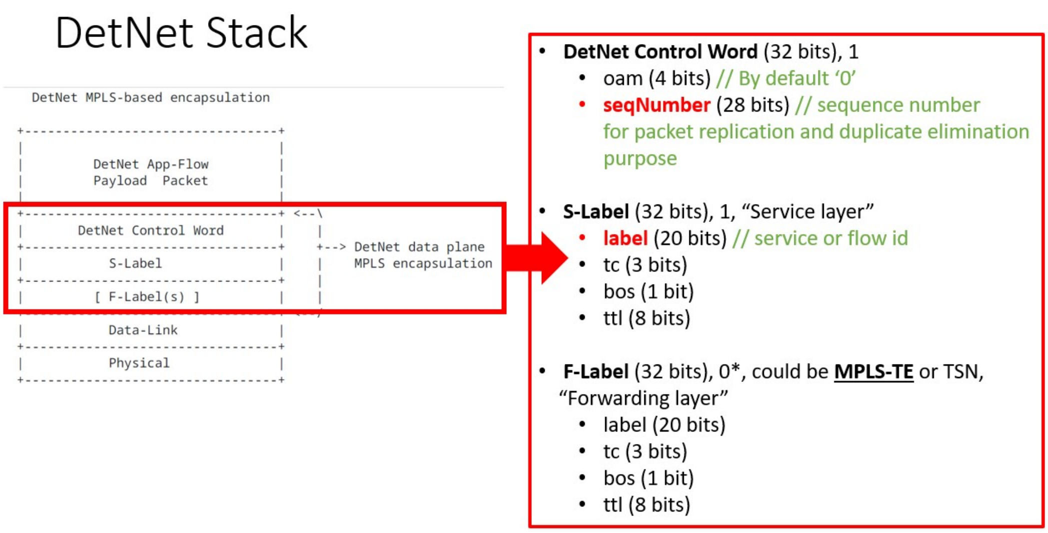

Figure 2. Example of DetNet MPLS-based encapsulation.

To assess the performance of FRER and PREOF, we are developing a testbed following the architecture defined in the following section.

2. TESTBED ARCHITECTURE

In this section, a three-node architecture is presented, featuring three activation modes to test the various techniques offered by TSN and DetNet for reducing packet loss in the network, see Table 1. For the sake of simplicity, we will refer to either Layer 2 Frames or Layer 3 Packets as packets in the rest of the paper. Additionally, two scenarios are outlined on this architecture, with one simulating the failure of a link in the network.

Table 1. Activation modes.

Mode 1	PREOF is activated
Mode 2	FRER is activated
Mode 3	Both FRER and PREOF are activated

2.1 Error-free scenario

In Figure 3, both the testbed architecture and an initial error-free scenario are depicted. This scenario consists of three nodes (Node 1, Node 2, and Node 3) and three links (Node 1 - Node 2, Node 1 - Node 3, and Node 2 - Node 3). Node 1 serves as the source node, while Node 3 serves as the destination. Each time a packet arrives at Node 1, it is replicated and forwarded through the links Node 1 - Node 2 and Node 1 - Node 3. Node 2 forwards any received packet to Node 3. At Node 3, if two identical packets arrive, one is discarded, and the other is forwarded.

With this initial scenario, we will be able to analyze the processing times of each node when dealing with TSN packets or DetNet packets, and observe the implications of header extension for packet identification in the case of TSN.

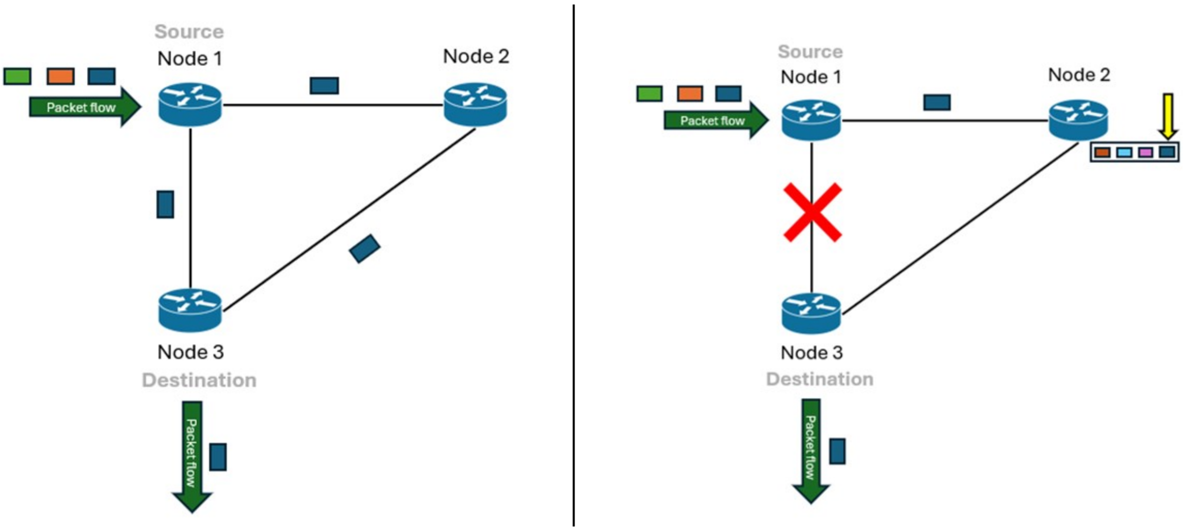

Figure 3. (Left) Error-free scenario. (Right) Error scenario.

2.2 Error scenario

We introduce a second scenario in which we simulate the failure of the link between Node 1 and Node 2, as shown in Figure 3. We elucidate the network behaviour in this second scenario across temporal intervals.

- T0. In Error scenario, the link between Node 1 and Node 2 is down. As a consequence of this state, Node 1 cannot send a packet A through that link to Node 3, thus the packet will only be sent through the Node 1 - Node 2 link. Simultaneously, this causes Node 2 to have a queue level nearing congestion, resulting in a longer processing time for a packet arriving at this node compared to the rest of the nodes in the network.

- T1. The previously downed link regains availability. Node 1 replicates and sends packet B through Node 1 - Node 2 and Node 1 - Node 3 links. Meanwhile, packet A continues to be processed at Node 2.

- T2. Packet B reaches Node 3 before packet A, which is still being processed at Node 2, causing an out-of-order delivery of packets at the destination.

- T3. Packet A arrives at Node 3.

The out-of-order delivery to the destination can potentially become a network issue and may even lead to failures at higher layers. For instance, if we are employing TCP as the protocol at layer 4, it implies the constraint of having an ordered delivery of packets to the destination. One potential solution in this scenario would be to implement ordering at Node 3. As mentioned at the beginning of this section, there are three activation modes in this architecture, but only in the modes where PREOF is activated will ordering be enabled at Node 3.

3. TESTBED IMPLEMENTATION

3.1 Initial test

Currently, in the Telefonica's laboratory, we have parallelly configured the aforementioned architecture for TSN with FRER, utilizing three Safran TSN-Z16 switches, and for DetNet with PREF (without ordering functionality) over MPLS-TE, through three EdgeCore CSP-7550 switches containing the Tofino chipset. The ordering functionality remains as future work, as it cannot be implemented solely with Tofino since an external buffer is required. Therefore, we have contemplated implementing this functionality through an FPGA.

At this stage, we are testing layer 2 capabilities, verifying that in TSN, an additional header is necessary to provide FRER. Figure 4 shows an example of how this encapsulation is performed on the TSN-Z16 equipment. As can be observed, a VLAN tag is added to Ethernet to enable operation within the TSN environment, and atop this, an additional R-TAG header is appended to define packet redundancy.

```
> Ethernet II, Src: SevenSolutio_20:8a:53 (64:fb:81:20:8a:53), Dst: aa:aa:aa:aa:aa:aa (aa:aa:aa:aa:aa:aa)
v 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 7
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0000 0111 = ID: 7
    Type: 802.1CB Frame Replication and Elimination for Reliability (0xf1c1)
v 802.1cb R-TAG
    <reserved>: 0x0000
    Sequence number: 5657
    Type: IPv4 (0x0800)
> Internet Protocol Version 4
```

Figure 4. TSN Encapsulation.

3.2 Work in progress

Both the TSN and DetNet architectures will be merged to enable DetNet over TSN. The ordering function will be added to DetNet through an FPGA. Additionally, a traffic generator (Spirent) connected to Node 1 will be incorporated into the architecture, along with instrumentation equipment introducing impairments (Paragon-X Calnex), which allows for simulating traffic congestion over Node 2.

4. CONCLUSIONS

In this paper, we conduct a comparative analysis of two techniques aimed at enhancing network reliability through deterministic solutions. These techniques are FRER for TSN and PREOF for DetNet. We have identified three significant differences between the two techniques: the layer level at which they operate, the utilization of an extension header for packet identification in the case of FRER, and the ordering functionality introduced by PREOF in DetNet. As part of future work, we will conduct various tests with the different activation modes on a DetNet/TSN architecture incorporating ordering functionality. The objective is to analyze the behavior of each technique and gather metrics related to latency times and packet losses across the network.

ACKNOWLEDGEMENTS

This work has been partially funded by the European Commission Horizon Europe SNS JU projects PREDICT-6G (GA 101095890) and DESIRE6G (GA 101096466), and the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union NextGenerationEU UNICO 5G I+D "Towards a smart and efficient telecom infrastructure meeting current and future industry needs" (TIMING) project (TSI-063000-2021-145, -148, -149).

REFERENCES

[1] B. Varga, J. Farkas, A. G. Malis, and S. Bryant, ‘Deterministic Networking (DetNet) Data Plane: IP over IEEE 802.1 Time-Sensitive Networking (TSN)’, Internet Engineering Task Force, Request for Comments RFC 9023, Jun. 2021. doi: 10.17487/RFC9023.

[2] ‘P802.1CBdb – FRER Extended Stream Identification Functions |’. Accessed: Apr. 17, 2024. [Online]. Available: https://1.ieee802.org/tsn/802-1cbdb/

[3] N. Finn, P. Thubert, B. Varga, and J. Farkas, ‘Deterministic Networking Architecture’, Internet Engineering Task Force, Request for Comments RFC 8655, Oct. 2019. doi: 10.17487/RFC8655.

[4] B. Varga, J. Farkas, S. Kehrer, and T. Heer, ‘Deterministic Networking (DetNet): Packet Ordering Function’, Internet Engineering Task Force, Request for Comments RFC 9550, Mar. 2024. doi: 10.17487/RFC9550.