Towards a Fault-Tolerant Architecture based on Time Sensitive Networking

Inés Álvarez, Manuel Barranco, Julián Proenza
Departament de Matemàtiques i Informàtica, Universitat de les Illes Balears, Spain
ines.Alvarez@uib.es, manuel.barranco@uib.es, julian.proenza@uib.es

This work is supported in part by the Spanish Agencia Estatal de Investigacin (AEI) and in part by FEDER funding through grant TEC2015-70313-R (AEI/FEDER, UE).

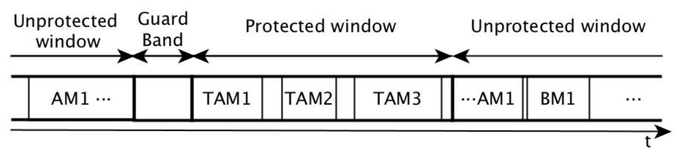
Abstract

The **Time Sensitive Networking (TSN) Task Group** has been working on describing a set of standards that will provide enhanced capabilities to **standard Ethernet.**
Specifically, they work to provide Ethernet with **real-time, reconfiguration** and **reliability** capacities. Nevertheless, this set of standards (commonly referred to as TSN) **does not cover some reliability aspects** that are relevant for the correct operation of **critical distributed control systems.**
Thus, in this work we present **a first proposal of a highly reliable architecture** and **a set of mechanisms based on TSN** to support the **real-time and reliability** requirements of these critical systems.

Time-Sensitive Networking Overview (I)

To achieve hard real-time **TSN** provides different standards that can be combined:

- P802.1 AS-rev: provides reliable time synchronisation for nodes and bridges.
- IEEE Std 802.1 Qbv: uses gates to control the transmission of frames and enforce a predefined schedule.
- IEEE Std 802.1 Qbu: pauses the transmission of low-priority frames to transmit higher-priority frames and then resumes the transmission of the paused frame.



Time-Sensitive Networking Overview (II)

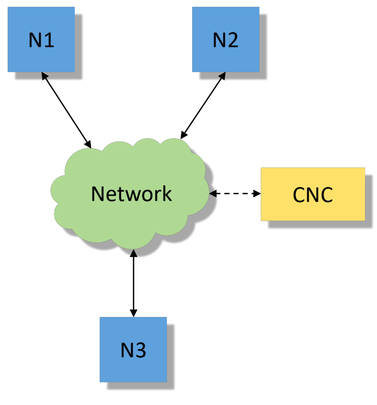
TSN relies on the **Stream Reservation Protocol (SRP)** to provide flexibility:

- **Real-time** flexibility: allows to register traffic of different classes with different real-time guarantees.
- **Operational** flexibility: allows to change the characteristics of the traffic on-line.

Devises the use of a central controller to calculate the new schedule and configure the network.

The communication is done through **streams,** that are requested by nodes and accepted or rejected by the CNC.

.



Time-Sensitive Networking Reliability

Three standards related to reliability:

- IEEE 802.1Qci: enables error containment. It allows to drop or assign new priorities to untimely frames or frames that exceed the bandwidth assigned to a given stream.
- IEEE 802.1Qca: describes new services, to allow for the creation of multiple and non-shortest paths between any pair of nodes and the further reservation of resources through those paths.
- IEEE 802.1CB: manages the replication of streams so one frame will be transmitted in parallel through each one of the multiple paths created by Qca. It defines how to **identify** streams that must be replicated, how frames should be **replicated at transmission** and **identified at reception**. Every element in the replicated paths, bridges and nodes, implement the replication and elimination mechanisms. This standard is called Frame Replication and Elimination for Reliability (FRER).

We propose a first design of a highly reliable architecture based on TSN standards, using a mono-hop architecture as a starting point.

Problem

TSN has several reliability standards, but does not cover all the aspects of a highly reliable network.

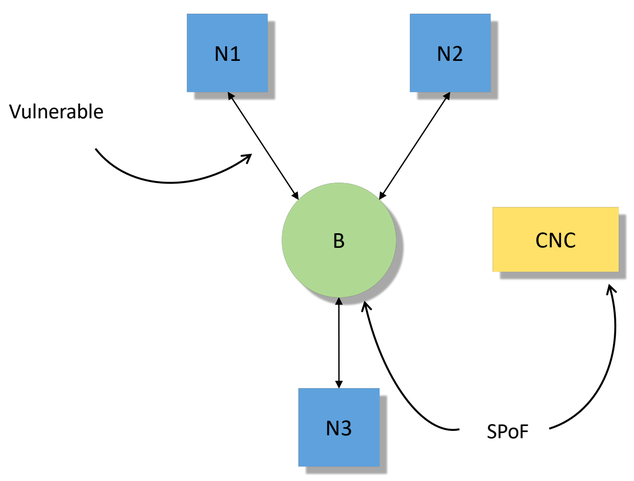
- The bridge and CNC are Single Points of Failure (SPoF).
- Links are vulnerable to temporary faults.
- Configuration mechanisms (requesting the creation of streams, deploying a schedule in the bridge...) are not reliable.

Which types of fault we want to tolerate?

- **Permanent** and **temporary** non-malicious operational hardware faults affecting the **network**.

How faults manifest?

- Bridges, CNCs and nodes exhibit byzantine failure semantics.
- Links exhibit omission failure semantics thanks to Ethernet's Frame Check Sequence.



Proposed Architecture

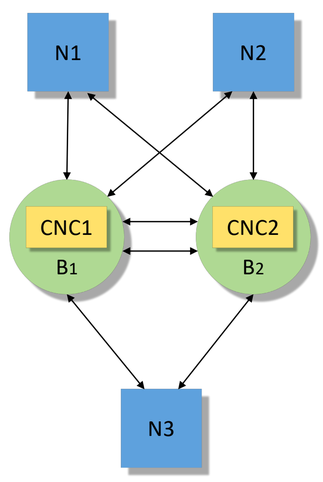
We propose a highly reliable architecture based on a mono-hop network and supported by TSN standards.

We restrict the failure semantics of bridges and CNCs to **crash** failure semantics. This can be done using internal duplication with comparison.

We restrict their failure semantics using error containment. Standard IEEE 802.1 Std **Qci** eliminates **timing** faults, but we need to **enhance** it to eliminate **impersonations** and **two-faced behaviours**.

Spatial replication of links can tolerate permanent faults. Interlinks allow CNCs to exchange configuration information and critical data. IEEE 802.1 Std Qca and CB can handle link redundancy.

P802.1 AS-Rev is devised to provide highly reliable synchronisation.



Replica determinism

To prevent inconsistencies both CNC-bridges must act as one; i.e. they must be **replica determinate**.
Replica determinism in bridges can be enforced with IEEE 802.1 As-Rev and IEEE 802.1 Qcv.
To guarantee it in the CNCs all the **configuration** must be done in lockstep with the same information:

1) CNCs must exchange their status of the network to ensure that the new configuration is done with the same information. This exchange is done through the interlinks.
2) To decide when to calculate the configuration we use the communication cycles. Every *n* cycles both CNCs exchange their status and choose the one with the highest amount of information. After that CNCs start calculating the new configuration.
3) To decide when the configuration is calculated we use the worst calculation time *w*. We assume that this time *w* is bounded and known in advance and that *w<n.* CNCs transmit their configuration in cycle *n+w*.

Conclusions and Future Work

Conclusions:

- TSN is a Task Group from IEEE that works to provide RT, reliability and flexibility to Ethernet.
- There are a series of standards finished and under development (commonly referred to as TSN).
- In this work we propose a first version of a highly reliable network architecture using TSN standards as the network technology.
- TSN must be extended to provide high reliability. We propose specific mechanisms to increase the reliability of the network and its mechanisms.
- We propose a technique to guarantee replica determinism.

Future work:

- Implement a prototype of the architecture and test the proposed mechanisms and techniques.