Simulation of the Proactive Transmission of Replicated Frames Mechanism over TSN

Inés Álvarez*, Drago Čavka**, Julián Proenza*, Manuel Barranco*

*Departament de Matemàtiques i Informàtica, Universitat de les Illes Balears, Spain

ines.alvarez@uib.es, julian.proenza@uib.es, manuel.barranco@uib.es

**Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina,

drago.cavka@etf.unibl.org

This work is supported in part by the Spanish Agencia Estatal de Investigación (AEI) and in part by FEDER funding through grant TEC2015-70313-R (AEI/FEDER, UE). Ines Alvarez and Drago Čavka was supported by a scholarship of the EUROWEB+ Project, which is funded by the Erasmus Mundus Action II programme of the European Commission.

Abstract

The Time-Sensitive Networking (TSN) Task Group (TG) is providing Ethernet with timing guarantees, reconfiguration services and fault tolerance mechanisms.
Some of TSN's targeted applications are real-time critical applications, which must provide a correct service continuously.
To support these applications the TSN TG standardised a spatial redundancy mechanism.
Even though spatial redundancy can tolerate permanent and temporary faults, it is not cost-effective.
Instead, temporary faults can be tolerated using time redundancy.
We proposed the Proactive Transmission of Replicated Frames (PTRF) mechanism to tolerate temporary faults in the links.
In this work we present a new PTRF approach, a PTRF simulation model and a comparison of the approaches using exhaustive fault injection.

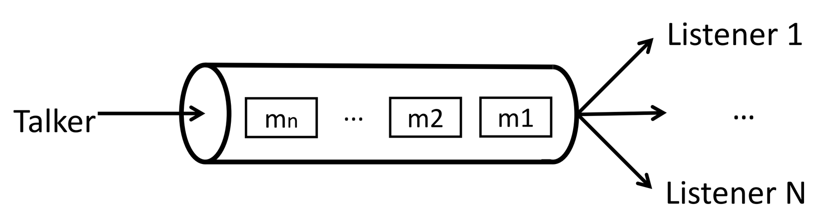
Time-Sensitive Networking Overview

TSN is a set of standards that aims at providing **Ethernet** with **hard real-time**, on-line **management** and **reliability** services.

To provide timing guarantees and enable on-line management of the network TSN relies, among others, on the SRP.

SRP enables the **reservation of resources along the path between two nodes** that want to communicate to guarantee availability and bounded transmission times.

The **communication is done through** virtual communication channels called **streams** and the resource reservation is done in a per-stream manner.

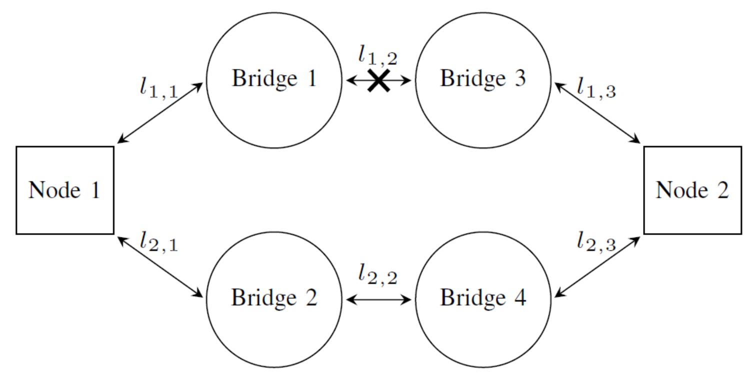


Problem

TSN **does not provide** any time-redundancy mechanisms in this level of the architecture specifically designed to **tolerate transient faults**. Although TSN can use **higher level protocols**, such as those based in **Automatic Repeat Request** (ARQ), this solution is not the best in real-time systems.

Using spatial redundancy to tolerate temporary faults is not adequate:

- The communication channel is specially vulnerable to transient faults.
- Spatial redundancy has high impact in the cost and size of the system.
- When permanent faults cause the attrition of the spatial redundancy, it may not be possible to tolerate transient faults any more.

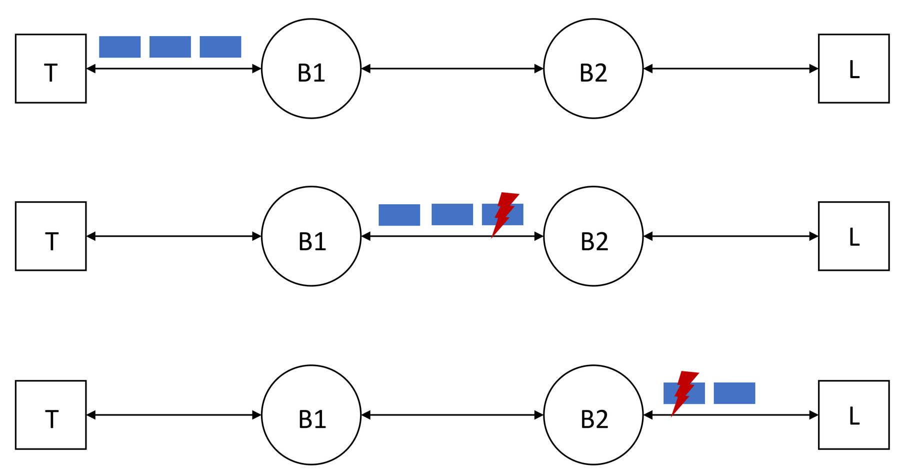


IN THIS WORK WE VALIDATE THE APPROACHES OF THE TIME REDUNDANCY MECHANISM THROUGH SIMULATION AND COMBINATION ANALYSIS

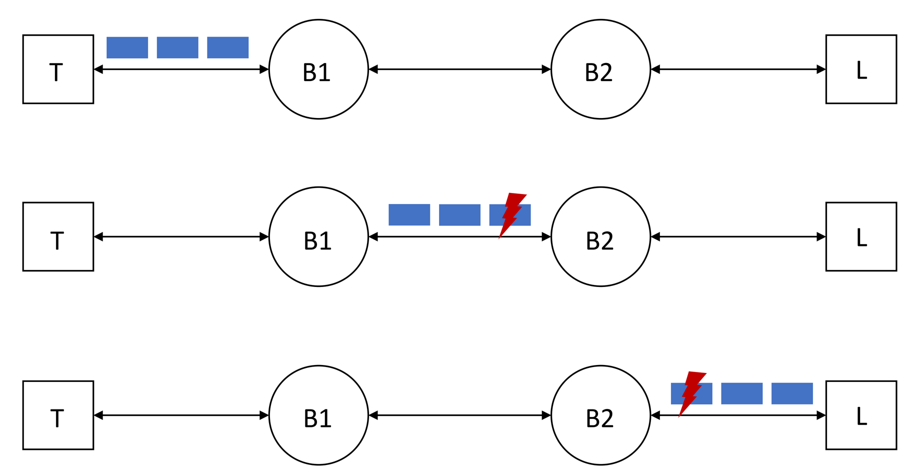
Proactive Time Redundancy

Use Proactive Transmission of Replicated Frames (PTRF) to tolerate temporary faults and TSN spatial redundancy to tolerate permanent faults in the links.

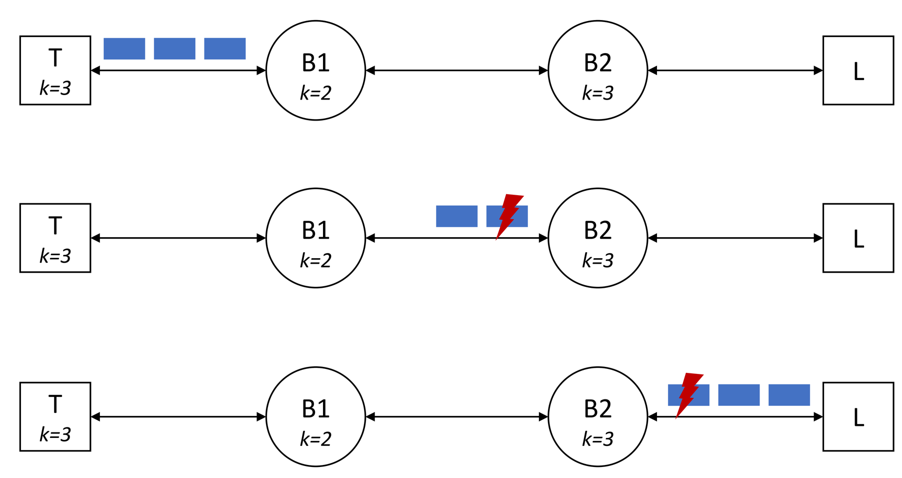
E2E estimation and replication of frames (A)



E2E estimation, link-based replication of frames (B)



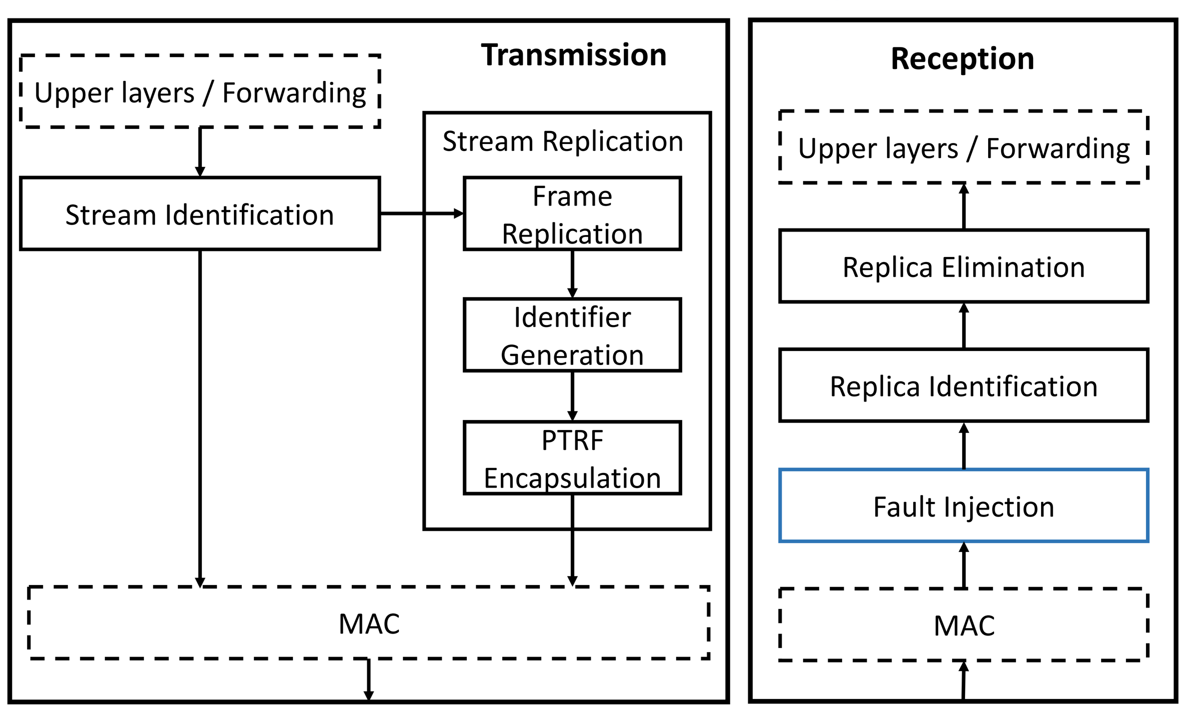
Link-based estimation and replication of frames (C)



OMNeT++ simulation model

We used simulation to evaluate and compare the proposed approaches.

Modules on transmission and reception



Frame Structure

Field	bytes
Destination MAC address	6
Source MAC address	6
C-tag EtherType	2
Priority, DE, VLAN ID	2
TRM EtherType	2
Message Identifier	2
Number of Replicas (k)	1
Payload Length/EtherType	2
data	n
Frame Check Sequence	4

Exhaustive fault injection

Inject all the possible combinations of errors where at least one replica traverses each link.

(A)

$$\sum_{e_1=0}^{k-1}\cdots\sum_{e_l=0}^{k-e_1-\cdots e_{l-1}-1}\left(\prod_{m=1}^{l}\binom{k-\sum_{i=1}^{m-1}e_i}{e_m}\right)$$

(B)

$$\left(\sum_{e'=0}^{k'-1}\binom{k'}{e'}\right)^{l}$$

(C)

$$\prod_{m=1}^{l}\sum_{e''=0}^{k''_m-1}\binom{k''_m}{e''}$$

$\boldsymbol{k, k', k''_m}$: number of replicas in the link
$\boldsymbol{e, e', e''_m}$: number of errors in the link
$\boldsymbol{l}$: number of links in the path

The goal of these experiments is twofold:

- Verify the correct operation of the approaches.
- Compare the approaches in terms of number of fault scenarios that can be tolerated.

The network parameters used are:

- 7 hops.
- 100 Mbps.
- No interfering traffic.

Approach	Links	Replicas	Tolerates scenarios
A	7	3	169
B	7	3	823543
C	7	2,2,2,3,3,4,4	297675

Results

We used the model to validate the design of the approaches. We injected all possible combinations of fault scenarios that can affect the replicas of a frame and we checked which ones are tolerated by each approach.

We also used the analyses to validate the results obtained with the model. The results obtained show that the number of fault scenarios tolerated by each approach during the simulations corresponds to the number obtained using the analyses.

We can conclude that it is feasible to build the approaches and that the design behaves as intended.

It is important to note that the actual reliability that is obtained with an approach is not directly proportional to the number of scenarios it tolerates. Even though tolerating a higher number of fault scenarios in this case is likely to improve reliability, the actual impact on the reliability also depends on the probability of each scenario. Approach A can tolerate a significant lower number of scenarios than approaches B and C, and we also see that reducing the number of replicas in approach C also impacts the number of scenarios.

Conclusions

We proposed the PTRF mechanism to tolerate temporary faults using proactive frame replication.
PTRF consists in transmitting several copies of each frame in a preventive manner to ensure that at least one copy reaches the destination even in the presence of temporary faults.
We proposed three approaches of this technique, of which the third one was presented in this work.
We developed a simulation model and used it to inject all combinations of fault scenarios to see which ones are tolerated by each approach.
We made a fault combination analysis to validate the results obtained with the simulation.

The results obtained with the simulation and the analysis were the same. This results allowed us to assess the feasibility of the three approaches. We saw that approach A can tolerate a lower number of fault scenarios than approaches B and C, and that the reduction in the number of replicas significantly impacts the number of tolerated scenarios.

Quantifying the actual reliability, as well as a real implementation of the mechanism is left as future work.