First Implementation and Test of a Node Replication Scheme on top of the Flexible Time-Triggered Replicated Star for Ethernet

MINISTERIO
DE ECONOMÍA
Y COMPETITIVIDAD

Fondo Europeo de Desarrollo Regional

EUROWEB
European Research and Education Collaboration with Western Balkan

Alberto Ballesteros
Sinisa Derasevic
David Gessner
Francisca Font
Inés Álvarez
Manuel Barranco
Julián Proenza

UIB
Universitat de les Illes Balears

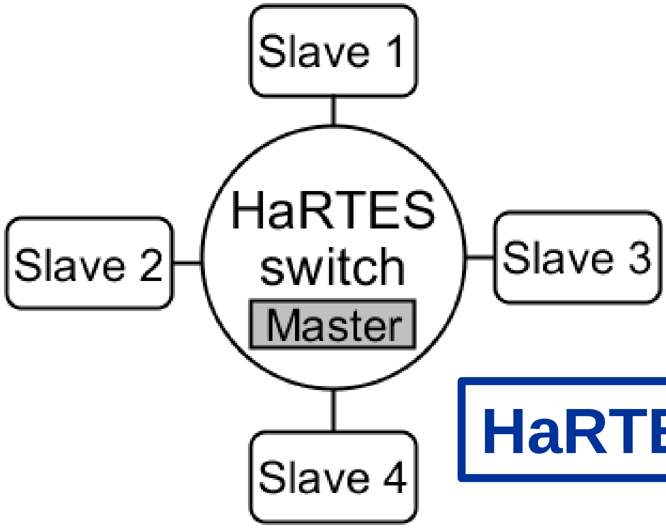
Introduction

The **FT4FTT** (Fault Tolerance for Flexible Time-Triggered Ethernet) project aims at providing an **architecture** that can support **distributed control applications** that are:

- **Predictable** → Have deadlines
- **Adaptive** → Have to work in changing environments
- **Highly-reliable** → Cannot suffer failures

FTT for Ethernet

Flexible Time-Triggered (FTT) on top of Ethernet allows developing **distributed embedded systems** that are **predictable** and **adaptive**

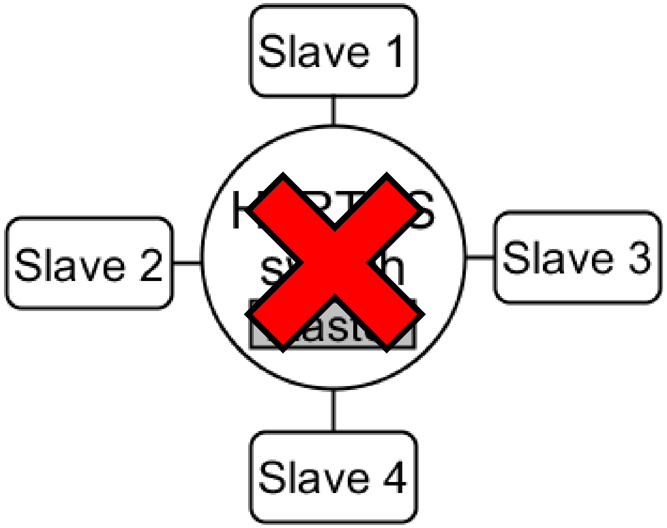


HaRTES

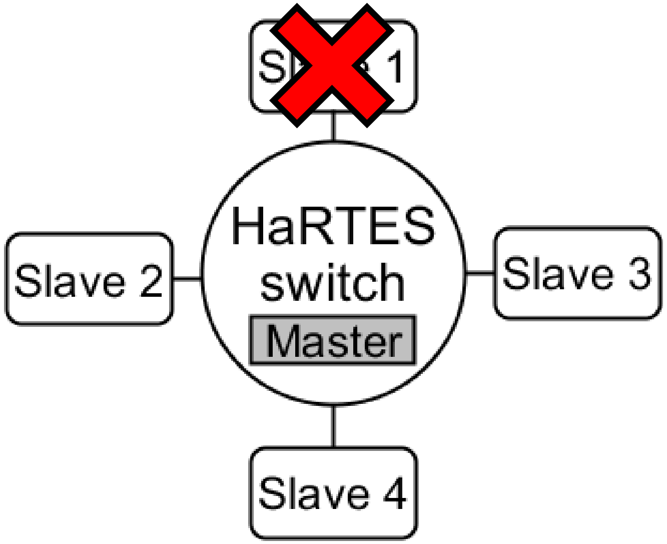
- **Master/multi-slave** comm. model
- **Slaves** are **regular nodes**
- The **master manages the comm.**

The FT4FTT project

Fault tolerance to faults affecting the **network**



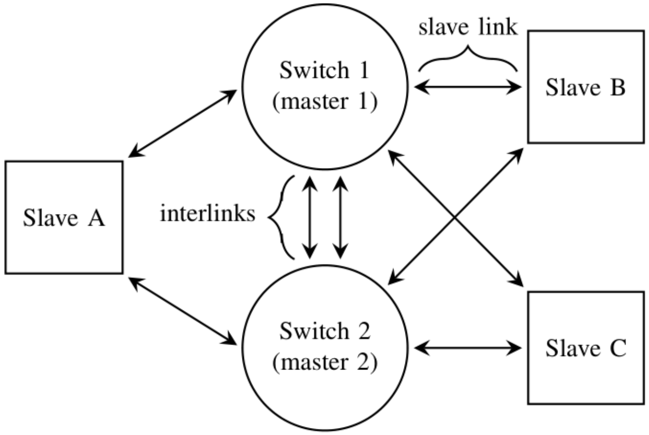
Fault tolerance to faults affecting the **nodes**



The FT4FTT project

Fault tolerance to faults affecting the **network**

- **Flexible Time-Triggered Replicated Star (FTTRS)**



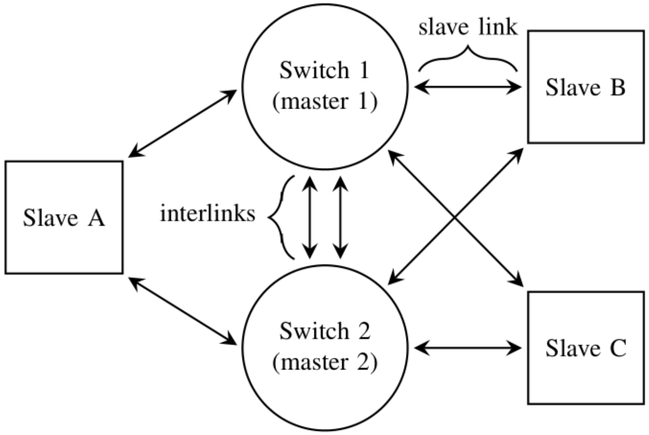
Fault tolerance to faults affecting the **nodes**

- Active replication
- Voting

The FT4FTT project

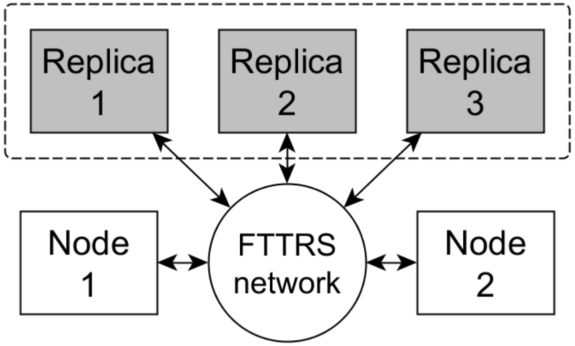
Fault tolerance to faults affecting the **network**

- Flexible Time-Triggered Replicated Star (FTTRS)



Fault tolerance to faults affecting the **nodes**

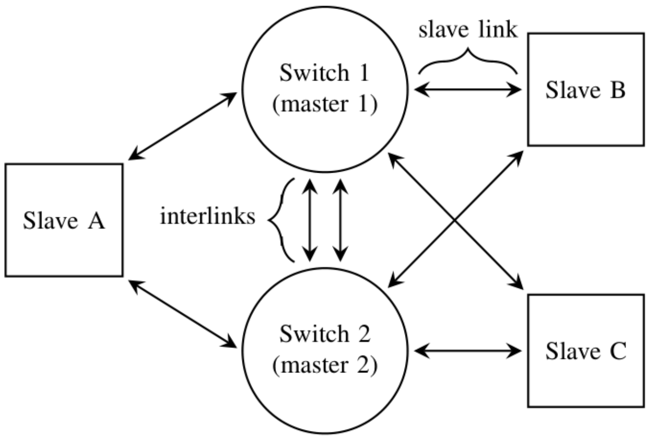
- **Active replication**
- Voting



The FT4FTT project

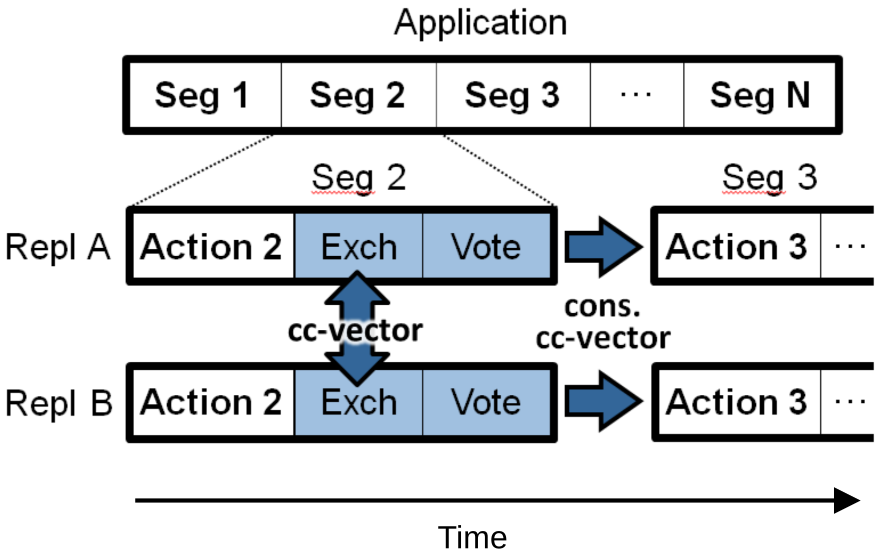
Fault tolerance to faults affecting the **network**

- Flexible Time-Triggered Replicated Star (FTTRS)



Fault tolerance to faults affecting the **nodes**

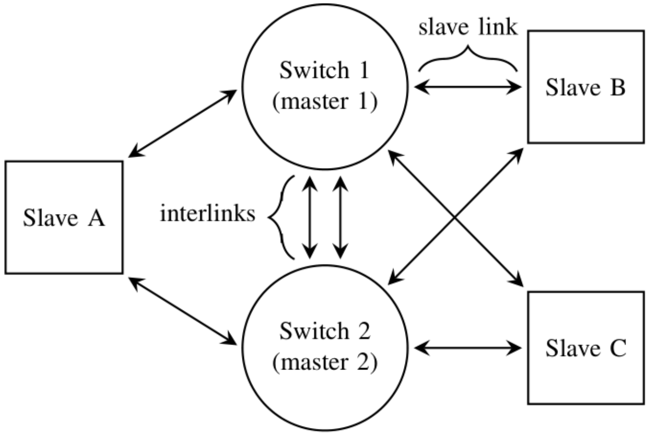
- Active replication
- **Voting**



The FT4FTT project

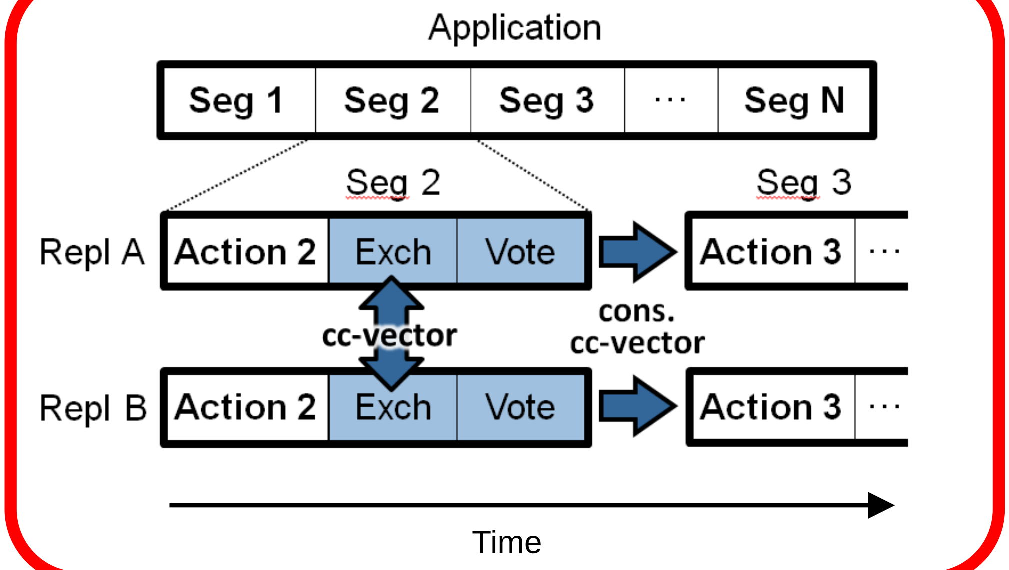
Fault tolerance to faults affecting the **network**

- Flexible Time-Triggered Replicated Star (FTTRS)



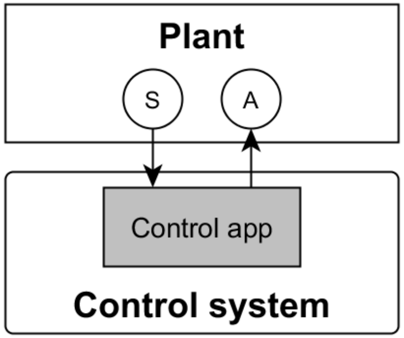
Fault tolerance to faults affecting the **nodes**

- Active replication
- **Voting**



The FT4FTT project

Typical **control applications** cyclically perform **three actions**: *sense*, *control* and *actuate*.



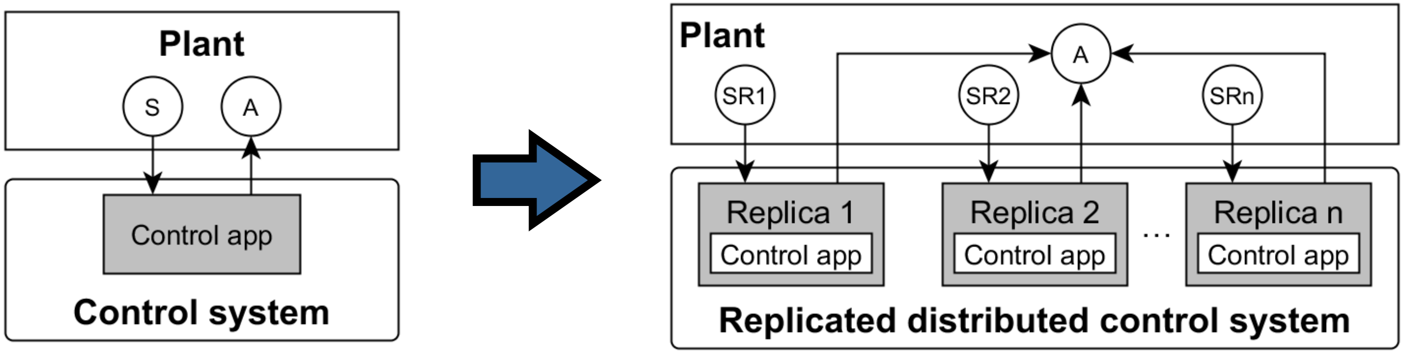
Sense

Ctrl.

Act.

The FT4FTT project

Typical **control applications** cyclically perform **three actions**: *sense*, *control* and *actuate*.



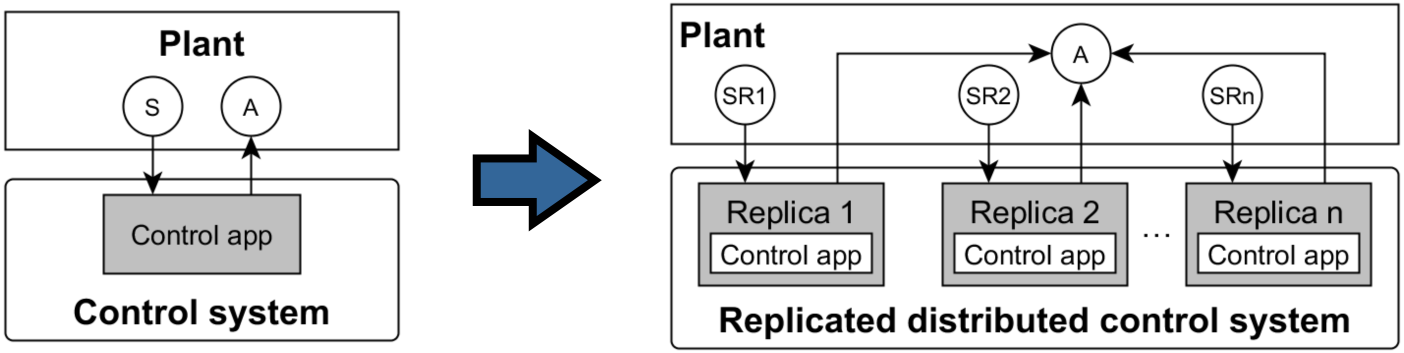
Sense

Ctrl.

Act.

The FT4FTT project

Typical **control applications** cyclically perform **three actions**: *sense*, *control* and *actuate*.



Plant

S

A

Control app

Control system

Plant

SR1

SR2

SRn

A

Replica 1

Control app

Replica 2

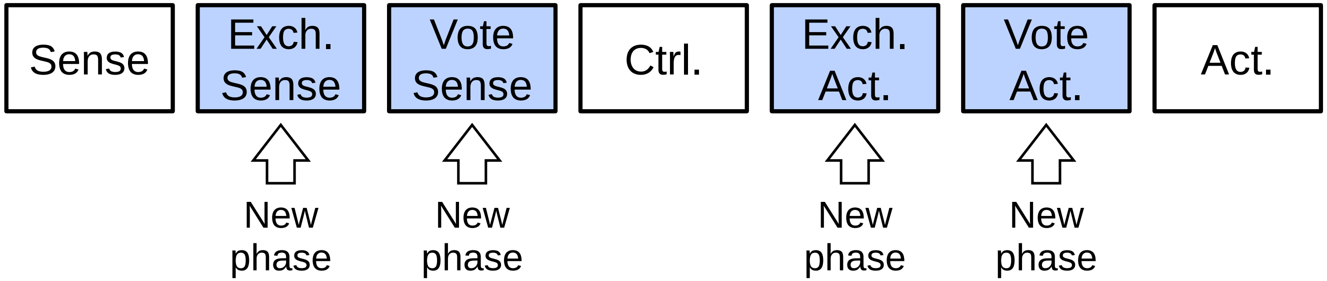
Control app

…

Replica n

Control app

Replicated distributed control system



Sense

Exch. Sense

Vote Sense

Ctrl.

Exch. Act.

Vote Act.

Act.

New phase

New phase

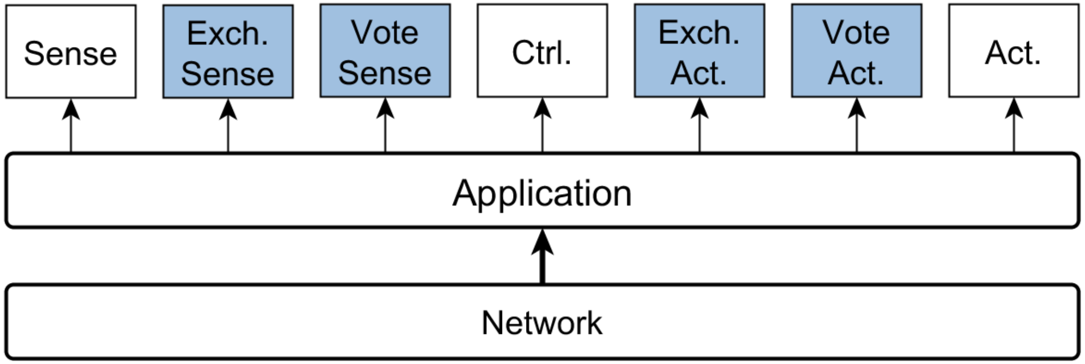
New phase

New phase

The CD4NR mechanism

In a previous work we designed the **Coordinate Dispatching for Node Replication** (CD4NR) mechanism to control the replicas:

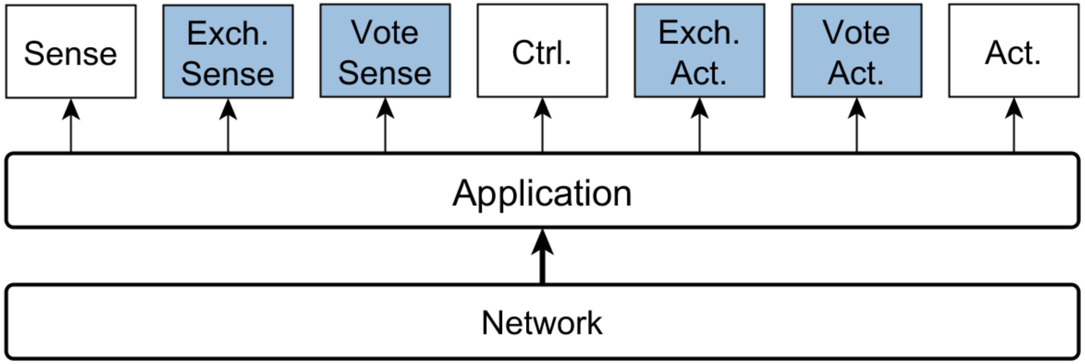
- **Operation** of the replicas
- **Transmission** of **messages**



The CD4NR mechanism

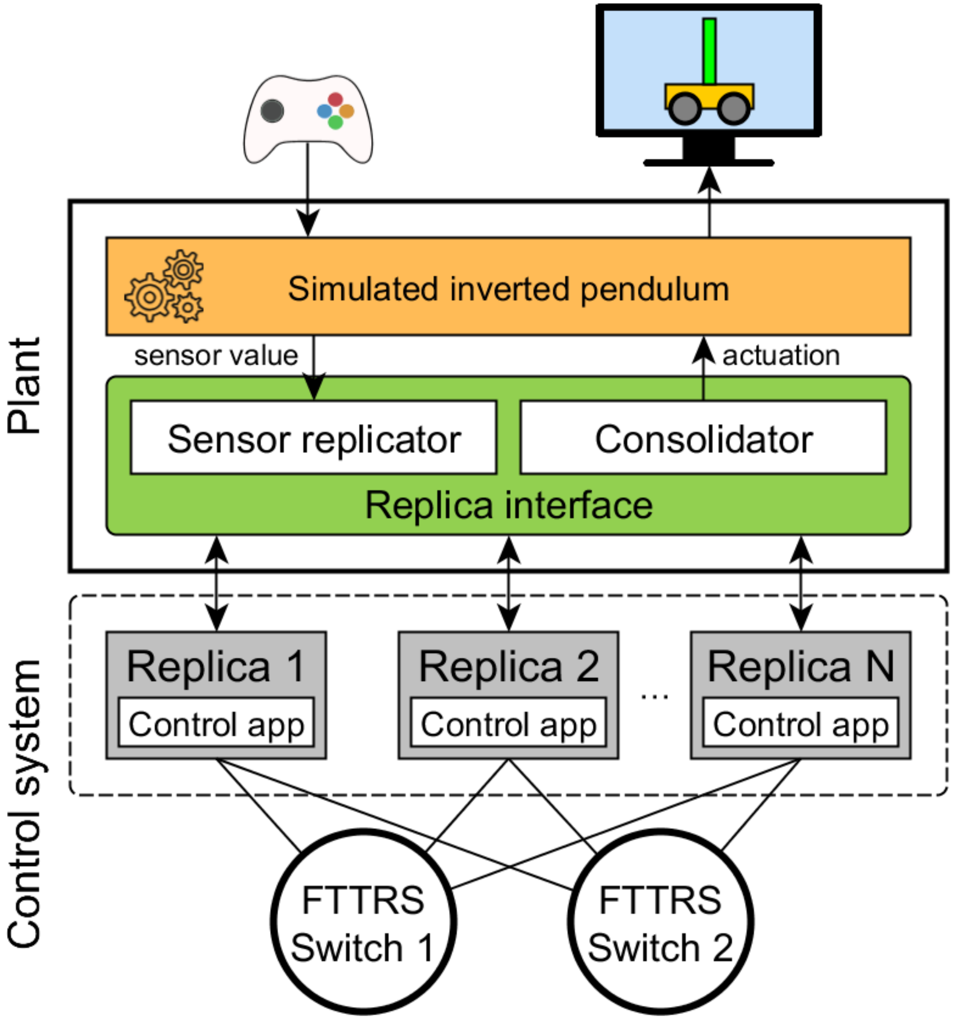
In a previous work we designed the **Coordinate Dispatching for Node Replication** (CD4NR) mechanism to control the replicas:

- **Operation** of the replicas
- **Transmission** of **messages**



In this work we present the **prototyping** and **testing** of the **CD4NR mechanism** and on a **real FTTRS network**

Implementation and testing



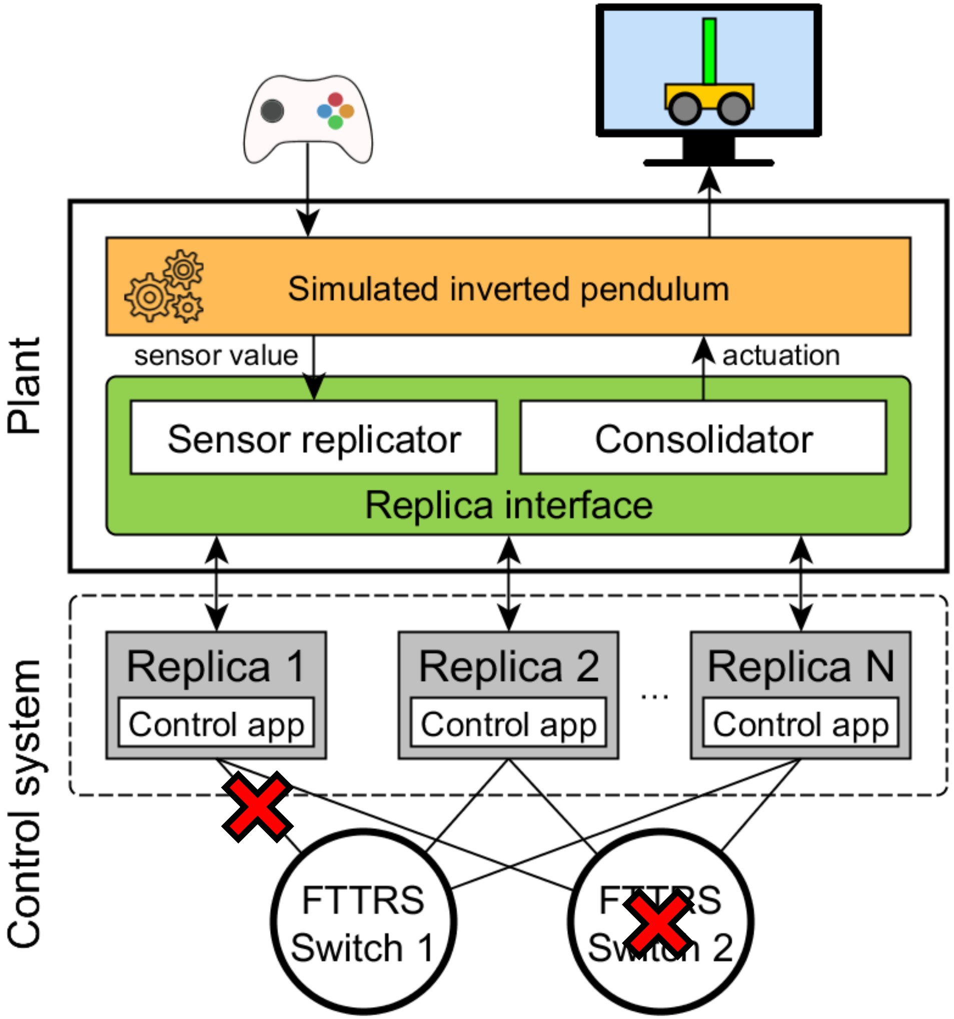
Implementation

- Modification of **replicas** and **switches**
- New **experimental setup**
  - Inverted pendulum
  - Hardware-in-the-loop

Testing

- **Validate** the mechanism
- **Verify** implementation and integration with the rest of mechanisms

Implementation and testing



Implementation

- Modification of **replicas** and **switches**
- New **experimental setup**
  - Inverted pendulum
  - Hardware-in-the-loop

Testing

Test the **tolerance** of the system to **permanent faults** affecting the **channel**

- Switch crashes
- Failures in the links

Testing

Validate the CD4NR mechanism, as

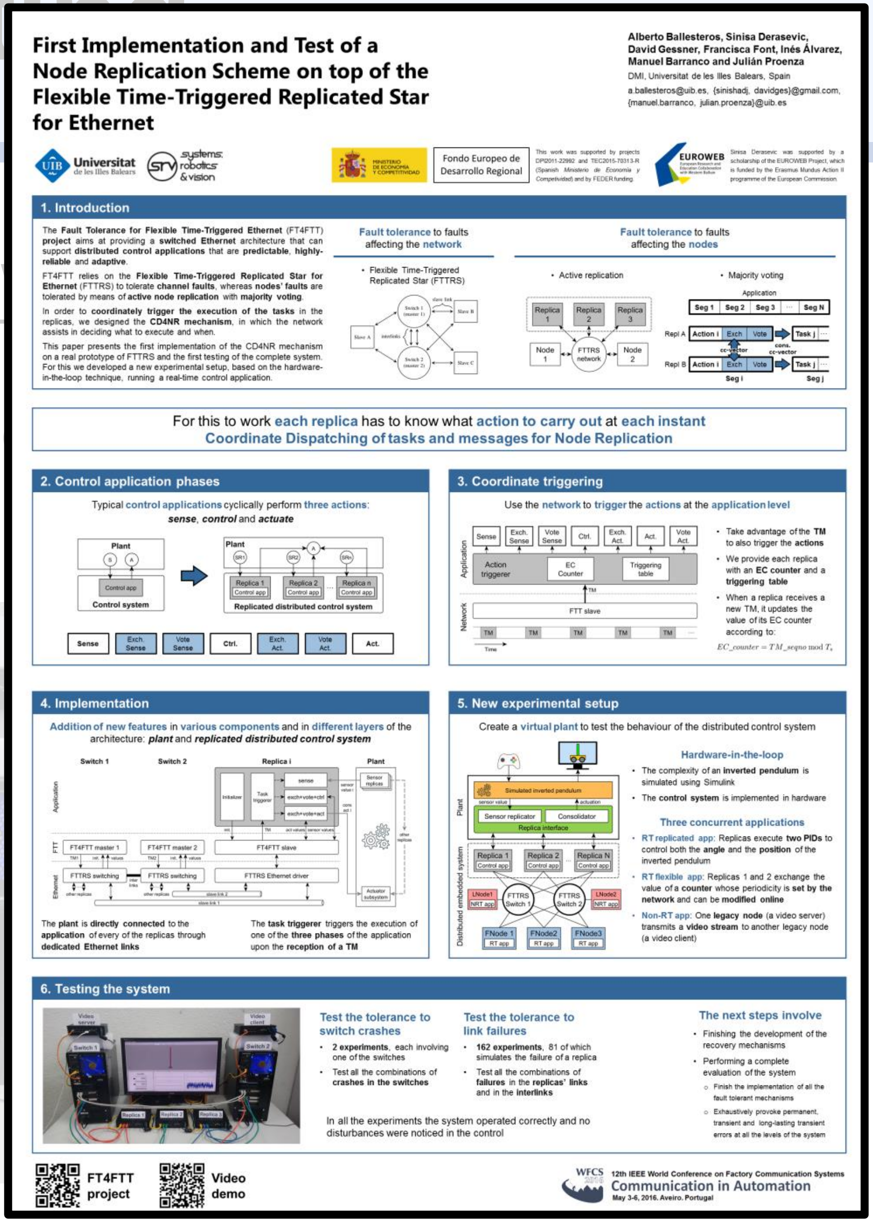
and integration with the rest of the FT

New experimental setup that implem

the hardware-in-the-loop technique

Thank you for your attention

See you at the poster session!



First Implementation and Test of a Node Replication Scheme on top of the Flexible Time-Triggered Replicated Star for Ethernet

Alberto Ballesteros, Sinisa Derasevic, David Gessner, Francisca Font, Inés Álvarez, Manuel Barranco and Julián Proenza

DMI, Universitat de les Illes Balears, Spain

a.ballesteros@uib.es, {sinishadj, davidges}@gmail.com, {manuel.barranco, julian.proenza}@uib.es

Universitat de les Illes Balears · systems, robotics & vision · Ministerio de Economía y Competitividad · Fondo Europeo de Desarrollo Regional

This work was supported by projects DPI2011-22992 and TEC2015-70313-R (Spanish *Ministerio de Economía y Competividad*) and by FEDER funding.

EUROWEB

Sinisa Derasevic was supported by a scholarship of the EUROWEB Project, which is funded by the Erasmus Mundus Action II programme of the European Commission.

1. Introduction

The **Fault Tolerance for Flexible Time-Triggered Ethernet** (FT4FTT) **project** aims at providing a **switched Ethernet** architecture that can support **distributed control applications** that are **predictable**, **highly-reliable** and **adaptive**.

FT4FTT relies on the **Flexible Time-Triggered Replicated Star for Ethernet** (FTTRS) to tolerate **channel faults**, whereas **nodes' faults** are tolerated by means of **active node replication** with **majority voting**.

In order to **coordinately trigger the execution of the tasks** in the replicas, we designed the **CD4NR mechanism**, in which the network assists in deciding what to execute and when.

This paper presents the first implementation of the CD4NR mechanism on a real prototype of FTTRS and the first testing of the complete system. For this we developed a new experimental setup, based on the hardware-in-the-loop technique, running a real-time control application.

Fault tolerance to faults affecting the network

- Flexible Time-Triggered Replicated Star (FTTRS)

Fault tolerance to faults affecting the nodes

- Active replication
- Majority voting

For this to work each replica has to know what action to carry out at each instant

Coordinate Dispatching of tasks and messages for Node Replication

2. Control application phases

Typical control applications cyclically perform three actions: ***sense***, ***control*** and ***actuate***

3. Coordinate triggering

Use the network to trigger the actions at the application level

- Take advantage of the **TM** to also trigger the **actions**
- We provide each replica with an **EC counter** and a **triggering table**
- When a replica receives a new TM, it updates the value of its EC counter according to:

$EC\_counter = TM\_seqno \bmod T_s$

4. Implementation

Addition of new features in various components and in different layers of the architecture: ***plant*** and ***replicated distributed control system***

The **plant** is **directly connected** to the **application** of every of the replicas through **dedicated Ethernet links**

The **task triggerer** triggers the execution of one of the **three phases** of the application upon the **reception of a TM**

5. New experimental setup

Create a virtual plant to test the behaviour of the distributed control system

Hardware-in-the-loop

- The complexity of an **inverted pendulum** is simulated using Simulink
- The **control system** is implemented in hardware

Three concurrent applications

- RT replicated app: Replicas execute **two PIDs** to control both the **angle** and the **position** of the inverted pendulum
- RT flexible app: Replicas 1 and 2 exchange the value of a **counter** whose periodicity is **set by the network** and can be **modified online**
- Non-RT app: One **legacy node** (a video server) transmits a **video stream** to another legacy node (a video client)

6. Testing the system

Test the tolerance to switch crashes

- **2 experiments**, each involving one of the switches
- Test all the combinations of **crashes in the switches**

Test the tolerance to link failures

- **162 experiments**, 81 of which simulates the failure of a replica
- Test all the combinations of **failures** in the **replicas' links** and in the **interlinks**

In all the experiments the system operated correctly and no disturbances were noticed in the control

The next steps involve

- Finishing the development of the recovery mechanisms
- Performing a complete evaluation of the system
  - Finish the implementation of all the fault tolerant mechanisms
  - Exhaustively provoke permanent, transient and long-lasting transient errors at all the levels of the system

FT4FTT project

Video demo

WFCS 2016 12th IEEE World Conference on Factory Communication Systems

Communication in Automation

May 3-6, 2016. Aveiro. Portugal

First Implementation and Test of a Node Replication Scheme on top of the Flexible Time-Triggered Replicated Star for Ethernet

MINISTERIO
DE ECONOMÍA
Y COMPETITIVIDAD

Fondo Europeo de
Desarrollo Regional

EUROWEB
European Research and
Education Collaboration
with Western Balkan

Alberto Ballesteros
Sinisa Derasevic
David Gessner
Francisca Font
Inés Álvarez

Manuel Barranco
Julián Proenza

UIB

Universitat de les Illes Balears