DFT4FTT: Dynamic FT for increasing the adaptivity of highly-reliable distributed embedded systems based on Flexible Time-Triggered Ethernet

Julián Proenza

Systems, Robotics and Vision Group. UIB.
SPAIN

Universitat de les Illes Balears

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD

DFT4FTT Project Data

- **DFT4FTT**
  Funded by the Spanish Gov. under grant TEC2015-70313-R
  - Part Spanish funding
  - Part FEDER funding

DFT4FTT Project Data

- **DFT4FTT**
  Funded by the Spanish Gov. under grant TEC2015-70313-R
    - Part Spanish funding
    - Part FEDER funding
  - 3-year project. **Started in Jan 2016 and ends in Dec 2018**
    - We can apply for an extension without extra funding.
  - Total money amount: 122.800,00 €
    - Funding for a technician (3 years) 80.300 €
    - Equipment 20.000 €
    - Travelling 20.000 €
    - Others (e.g. journal publication costs)
  - Research team (doctors teaching at the UIB)
  - Work team (foreign doctors and other personnel)

The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short

The current team

- **UIB**
  - **Manuel Barranco**
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short

The current team

- **UIB**
  - Manuel Barranco
  - **Ignasi Furió**
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short

The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - **Pere Palmer**
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short



The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - **David Gessner**
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short




The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - **Sinisa Djerasevic**
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short





The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - **Alberto Ballesteros** (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short






The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - **Inés Álvarez** (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short







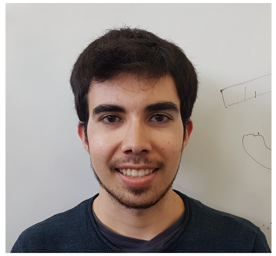
The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - **Daniel Bujosa** (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short

The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - **Sergi Arguimbau** (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short







The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short

The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza

- **MDH**
  - **Guillermo Rodríguez-Navas**
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short

The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza


- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - **Luís Almeida**
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - Michael Short

The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - **Paulo Pedreiras**
- **Teesside Univ. (UK)**
  - Michael Short



The current team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - Inés Álvarez (PhD thesis)
  - Daniel Bujosa (part-time technician)
  - Sergi Arguimbau (part-time technician)
  - Julián Proenza
- **MDH**
  - Guillermo Rodríguez-Navas
- **UPorto**
  - Luís Almeida
- **UAveiro**
  - Paulo Pedreiras
- **Teesside Univ. (UK)**
  - **Michael Short**




Collaborators

- **UIB**
  - Yolanda González
  - Patricia Arguimbau
- **University of Banja Luka (Bosnia and Herzegovina)**
  - Drago Cavka

Collaborators

- **UIB**
  - **Yolanda González**
  - Patricia Arguimbau

- **University of Banja Luka (Bosnia and Herzegovina)**
  - Drago Cavka

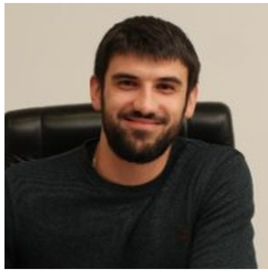
Collaborators

- **UIB**
  - Yolanda González
  - **Patricia Arguimbau**

- **University of Banja Luka (Bosnia and Herzegovina)**
  - Drago Cavka

Collaborators

- **UIB**
  - Yolanda González
  - Patricia Arguimbau
- **University of Banja Luka (Bosnia and Herzegovina)**
  - **Drago Cavka**



Context of the project

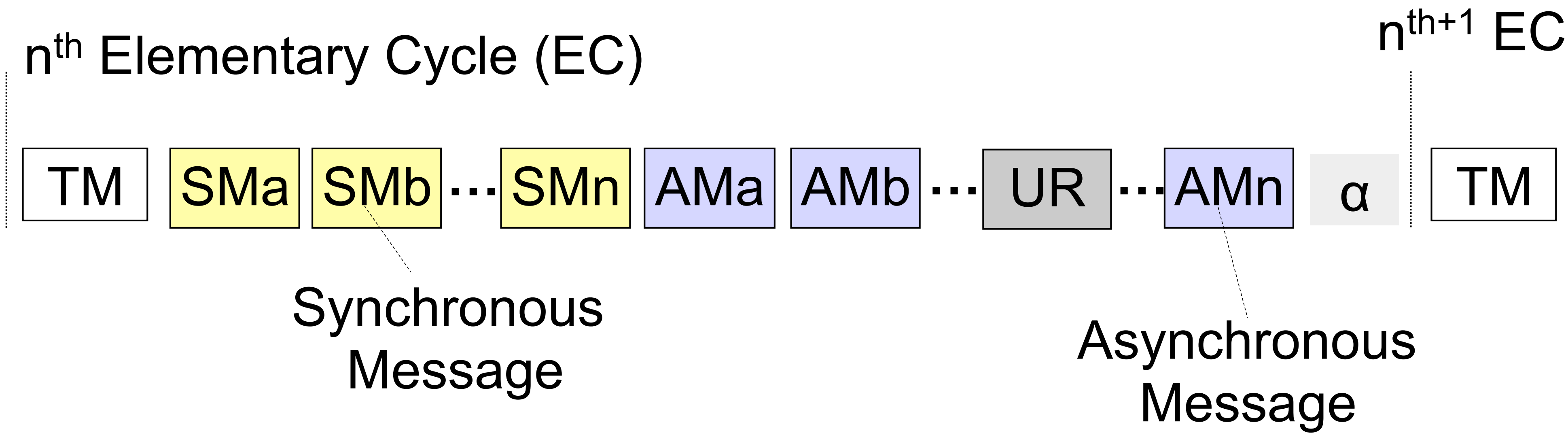
- Many embedded systems have strict requirements on **real-time performance** and **dependability.**
- The current tendency is to apply embedded systems also in **dynamic environments**
  - operating conditions may change frequently and in an unpredictable manner.
- Such systems are called **adaptive embedded systems**, and require services supporting…
  - flexibility, real-time and dependability at different levels of the system architecture, such as the OS and **the network**.

Context of the project
Flexibility in FTT

- **FTT is a very adequate networking paradigm** for developing adaptive distributed embedded systems,
  - developed in U. Aveiro (Portugal)
  - it already provides certain communication services that are very well suited for adaptivity, **i.e. flexibility in the real-time response**

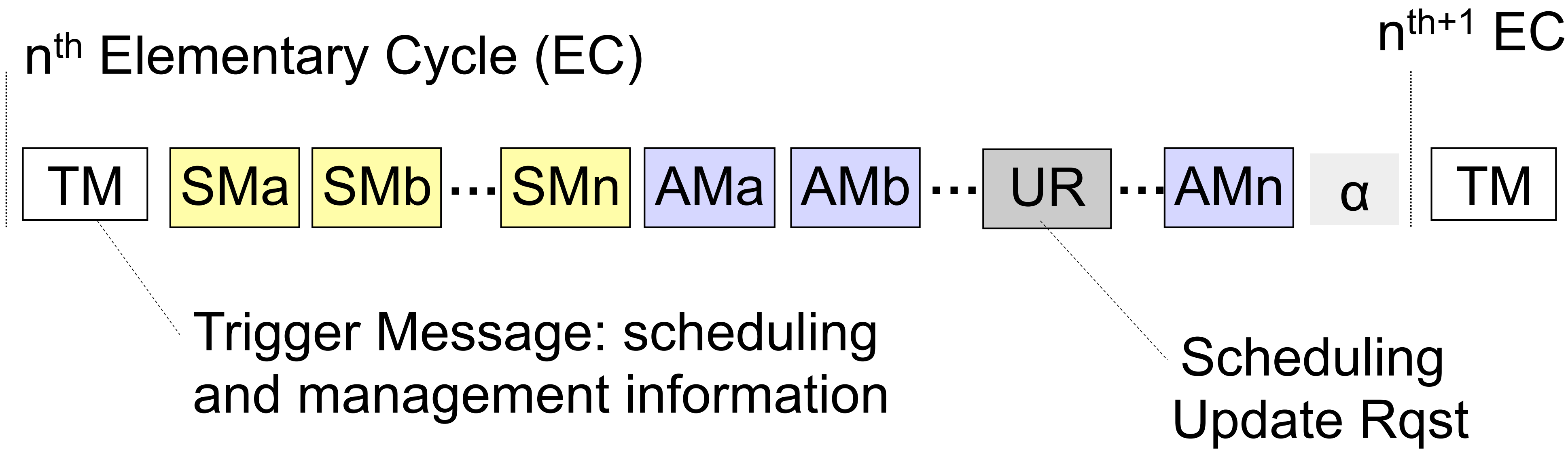
Context of the project
Flexibility in FTT

- **First**, FTT is able to **convey different types of traffic**: time is divided in Elementary Cycles (ECs) and each EC in a synchronous and an asynchronous window



Context of the project Flexibility in FTT

- **Second**, FTT is able to **dynamically change its real-time response**: nodes can request changes in the messages to be sent in real-time and a **master** decides if each request is **schedulable**.

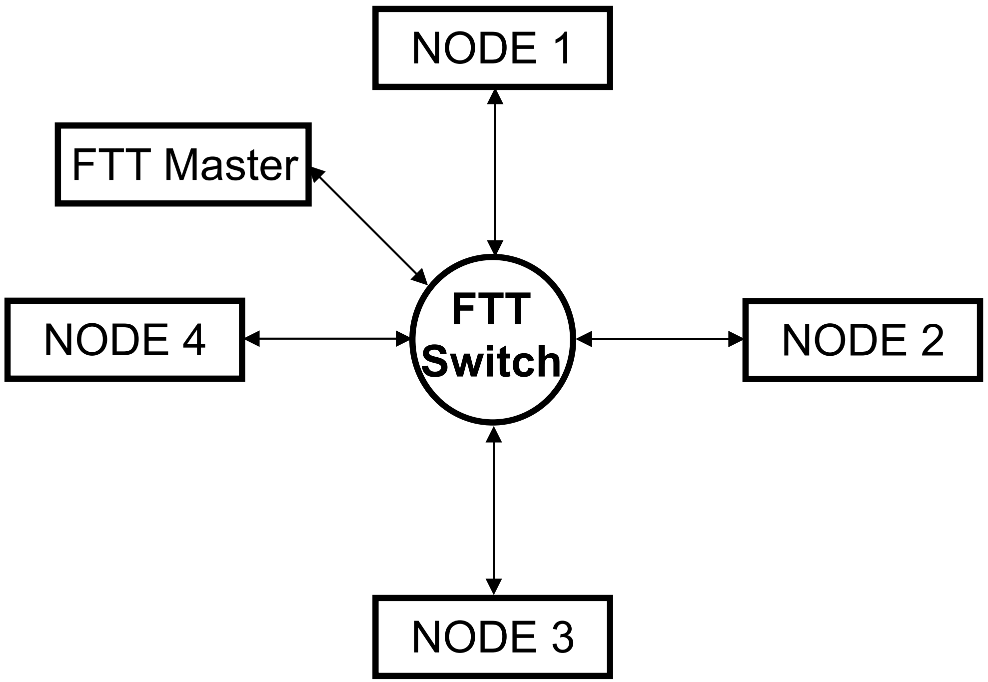


Context of the project FTT-Ethernet

- **FTT-Ethernet** is the result of using FTT over the appealing Ethernet with RT response,
  - A higher potential thanks to the increase in bandwidth
  - **An FTT Switch (HaRTES) allows using legacy nodes**

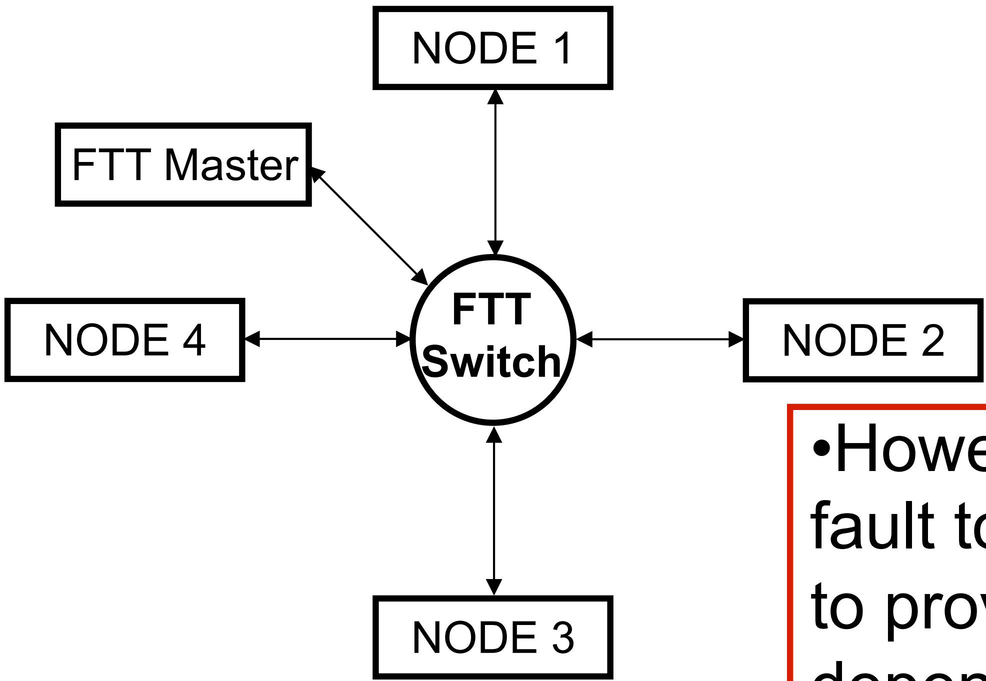
Context of the project FTT-Ethernet

- **FTT-Ethernet** is the result of using FTT over the appealing Ethernet with RT response,
  - A higher potential thanks to the increase in bandwidth
  - **An FTT Switch (HaRTES) allows using legacy nodes**



Context of the project FTT-Ethernet

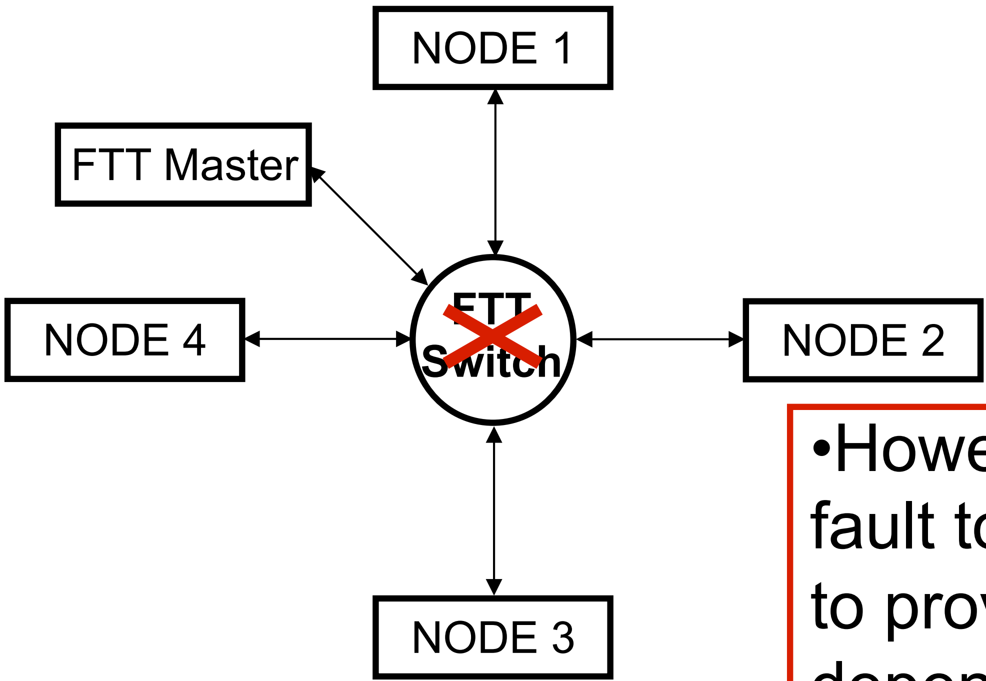
- **FTT-Ethernet** is the result of using FTT over the appealing Ethernet with RT response,
  - A higher potential thanks to the increase in bandwidth
  - **An FTT Switch (HaRTES) allows using legacy nodes**



•However, it lacks the fault tolerance features to provide the desired dependability

Context of the project FTT-Ethernet

- **FTT-Ethernet** is the result of using FTT over the appealing Ethernet with RT response,
  - A higher potential thanks to the increase in bandwidth
  - **An FTT Switch (HaRTES) allows using legacy nodes**



•However, it lacks the fault tolerance features to provide the desired dependability

Motivation of the **(previous)** FT4FTT project

- Solving this limitation of FTT-Ethernet would represent a significant step forward in the **development of the future adaptive distributed embedded systems**.

Goal of the FT4FTT project

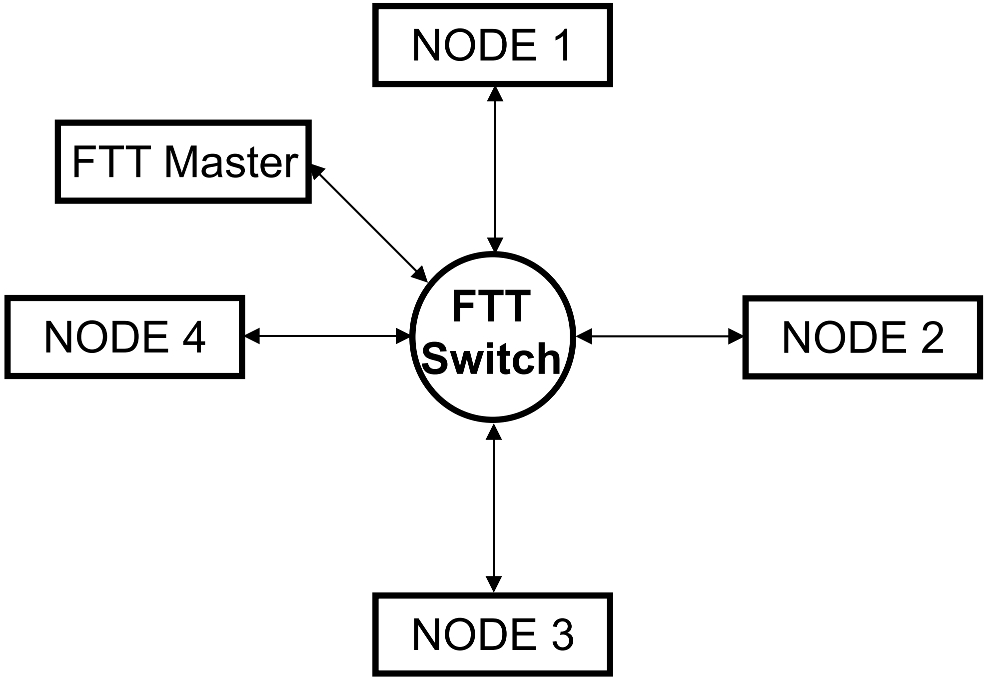
- **The design, implementation and validation of a highly-dependable communication infrastructure based on FTT-Ethernet.**

Goal of the FT4FTT project

- **The design, implementation and validation of a highly-dependable communication infrastructure based on FTT-Ethernet.**
  - … by adding Fault Tolerance Mechanisms

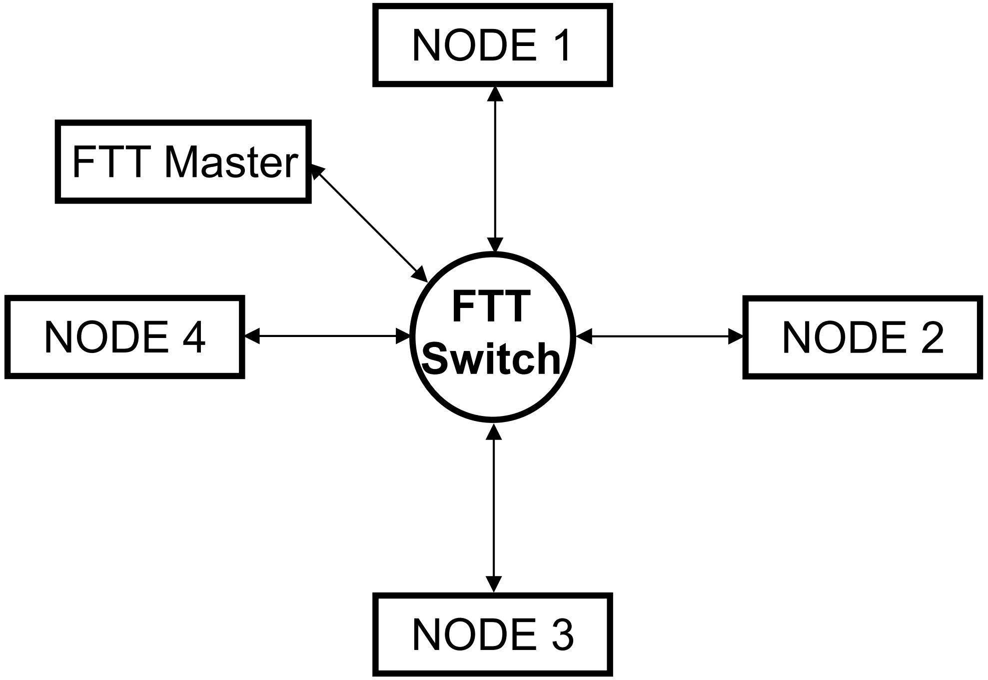
Our starting point

- An FTT-Ethernet network
  - actually HaRTES was in the initial proposal



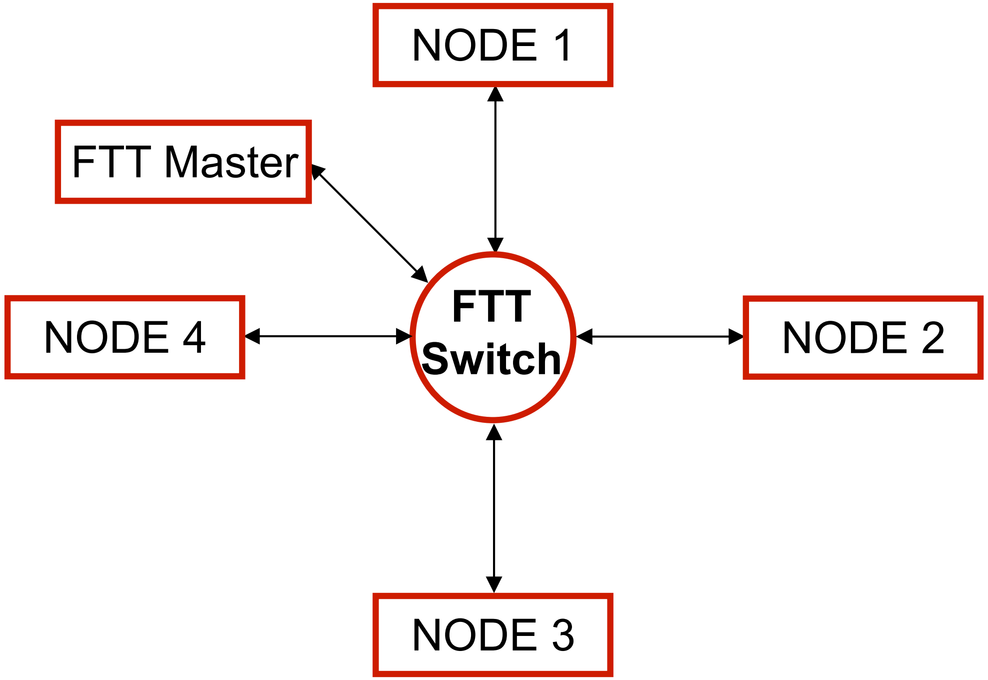
Our approach

- To design **a complete FT system**
  - since dependability is a property that has to be guaranteed in the system as a whole



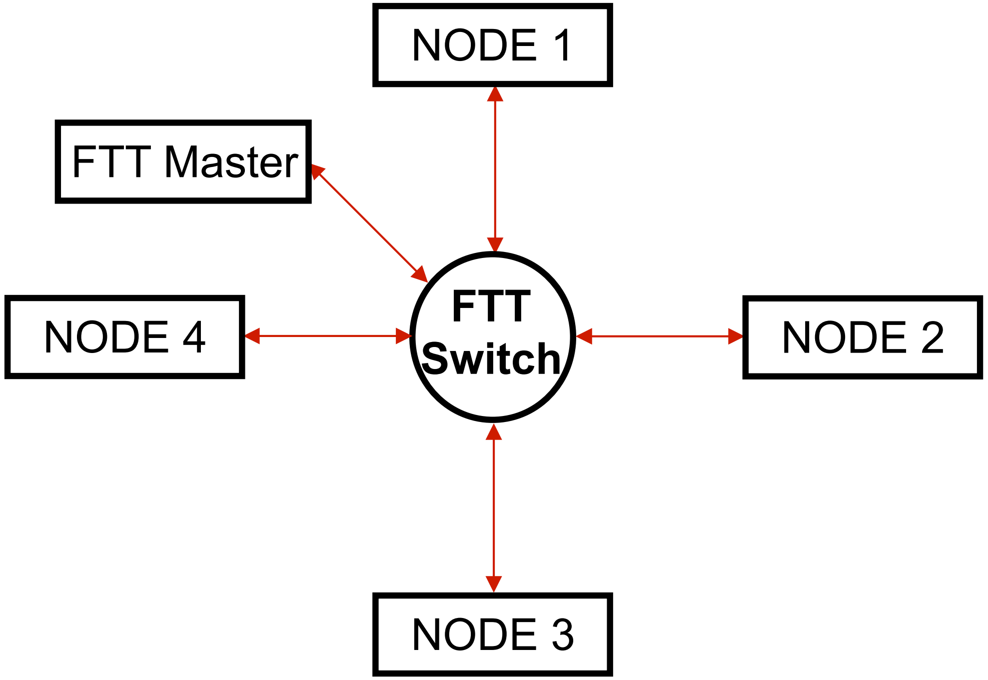
Fault model (1)

- Permanent and temporary faults in the HW modules
  - Slaves
  - Masters
  - Switches



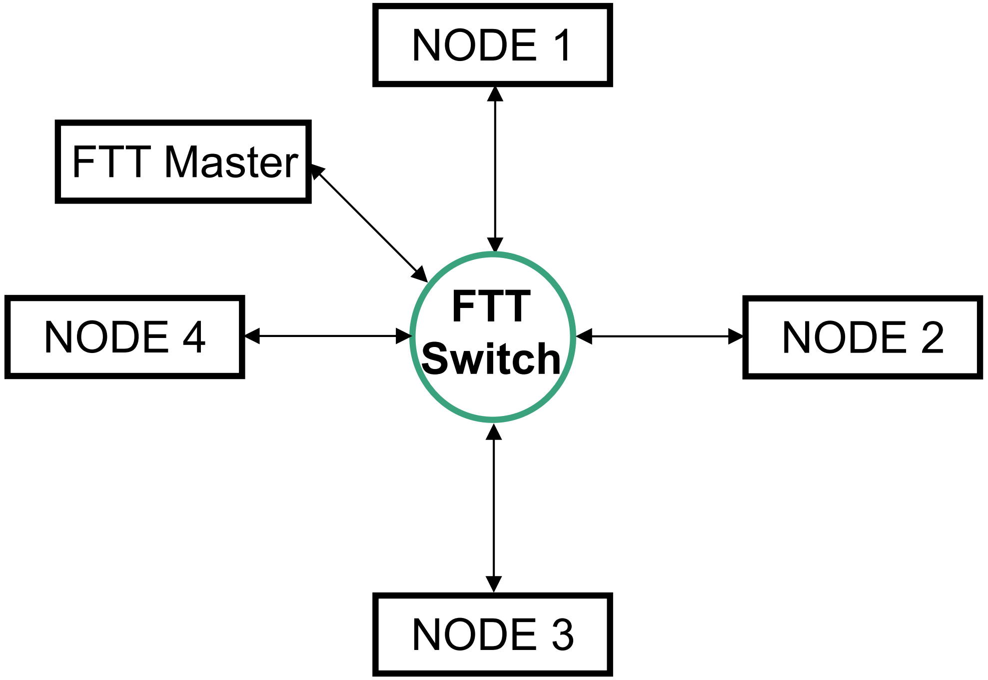
Fault model (2)

- Permanent and temporary faults in the links



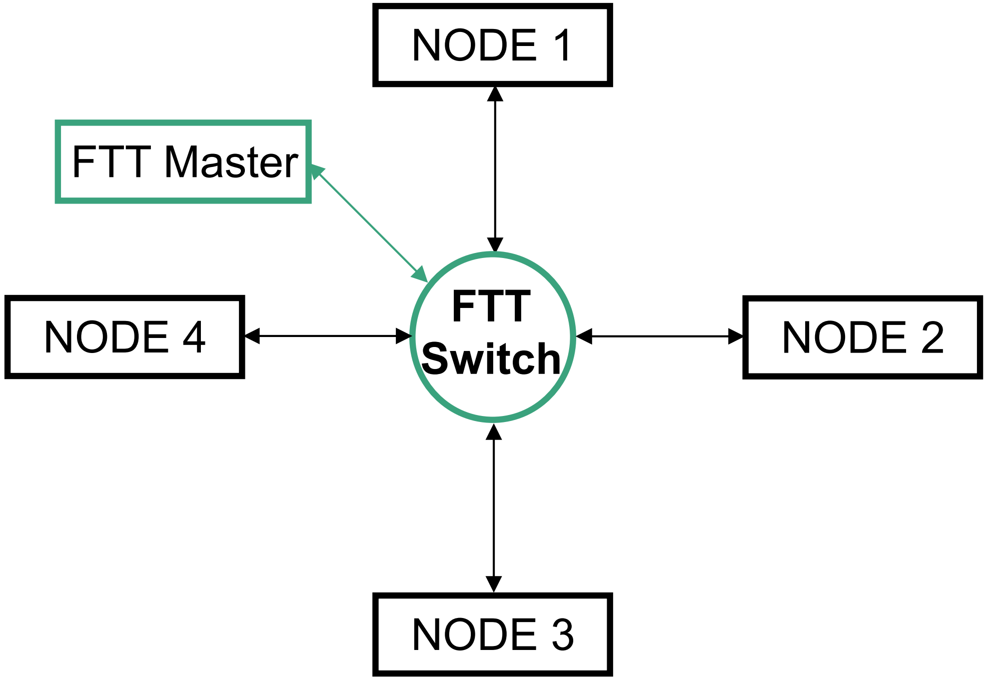
Our strategy (1)

- Follow as much as possible the current FTT strategy of **concentrating most of the additions in the switches**
  - to be able to work with **COTS nodes and even legacy nodes**
  - to have a direct link between master and port guardians



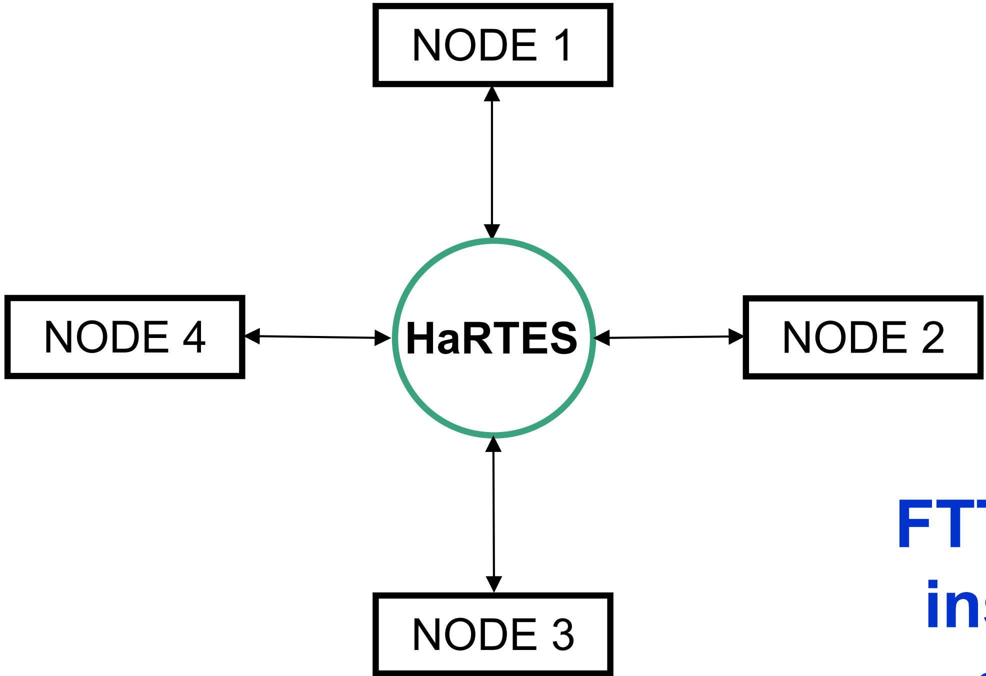
Our strategy (2)

- It is not just to add FT to FTT but also **to make the most of the FTT features in order to simplify/improve the FT mechanisms** that need to be added



Building the Fault-Tolerant System

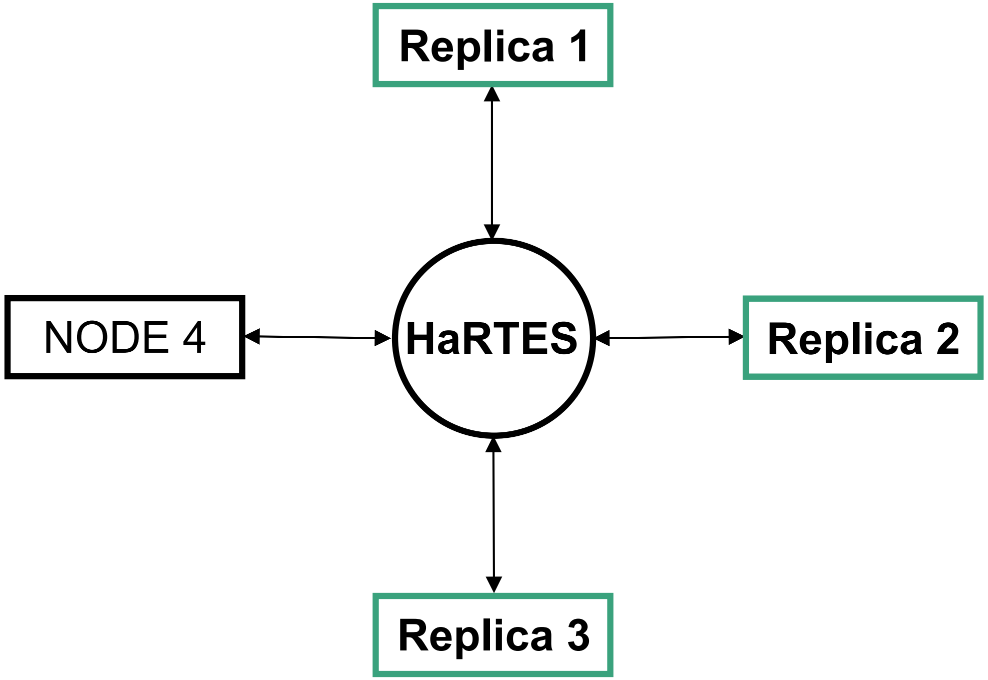
- Our choice was to **take HaRTES as corner-stone**
  - It is a **custom Ethernet switch with FTT built-in**



FTT master inside the switch

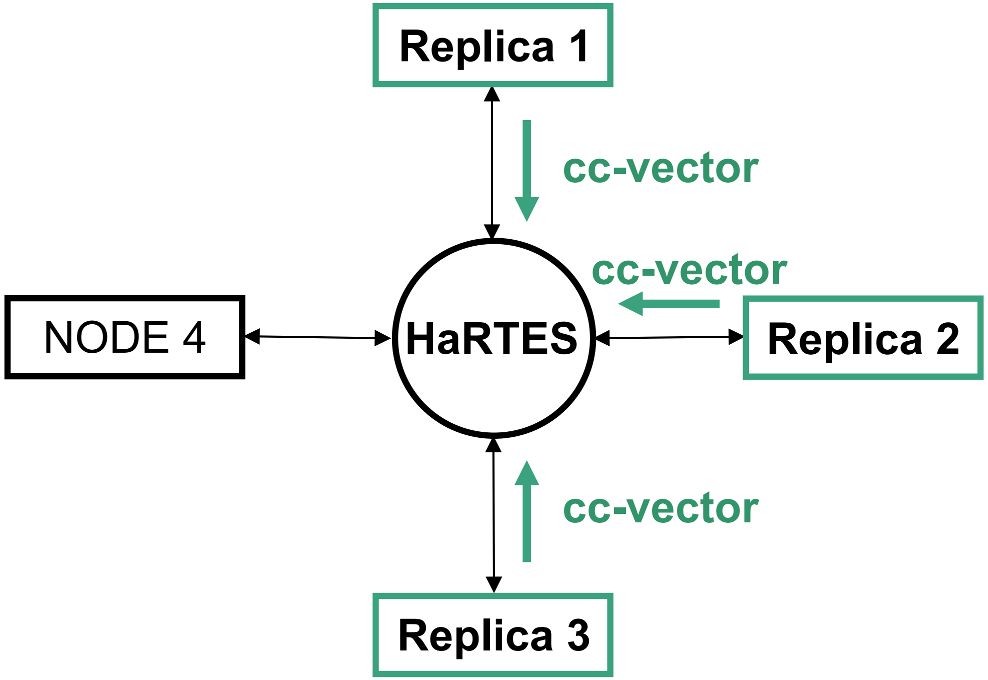
Node (FTT slave) Replication

- **Tolerating node faults is mandatory for high reliability**
  - We use **active replication** for the slaves



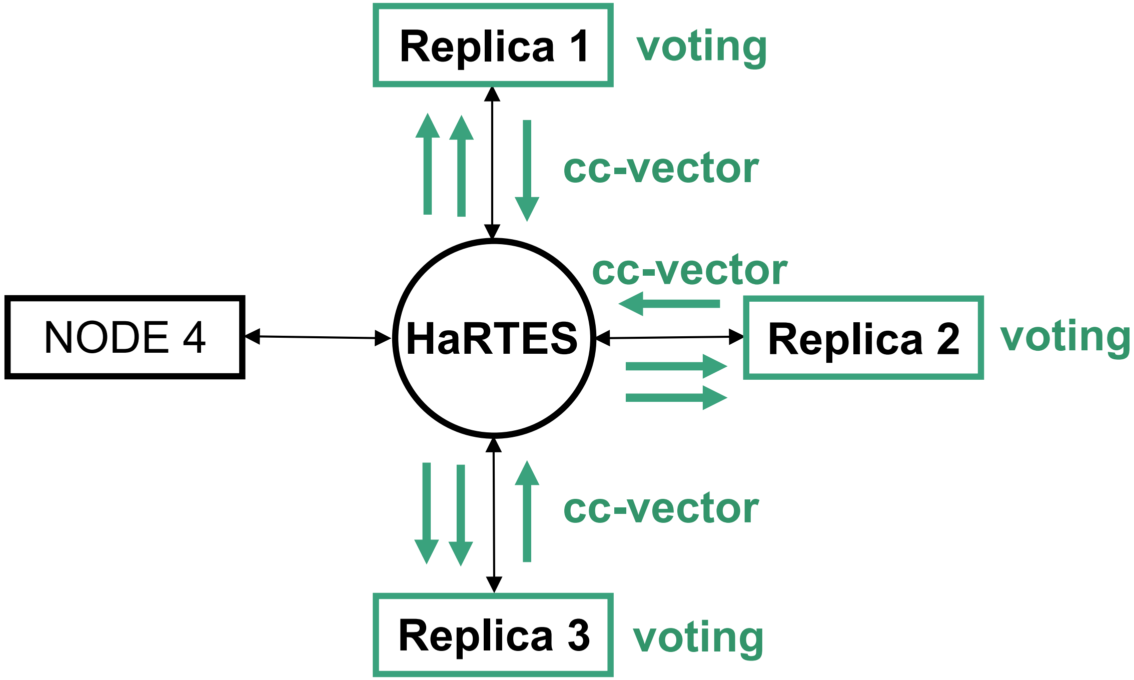
Node (FTT slave) Replication

- **active** replication for the slaves
  - using **N-Version Programming** terminology



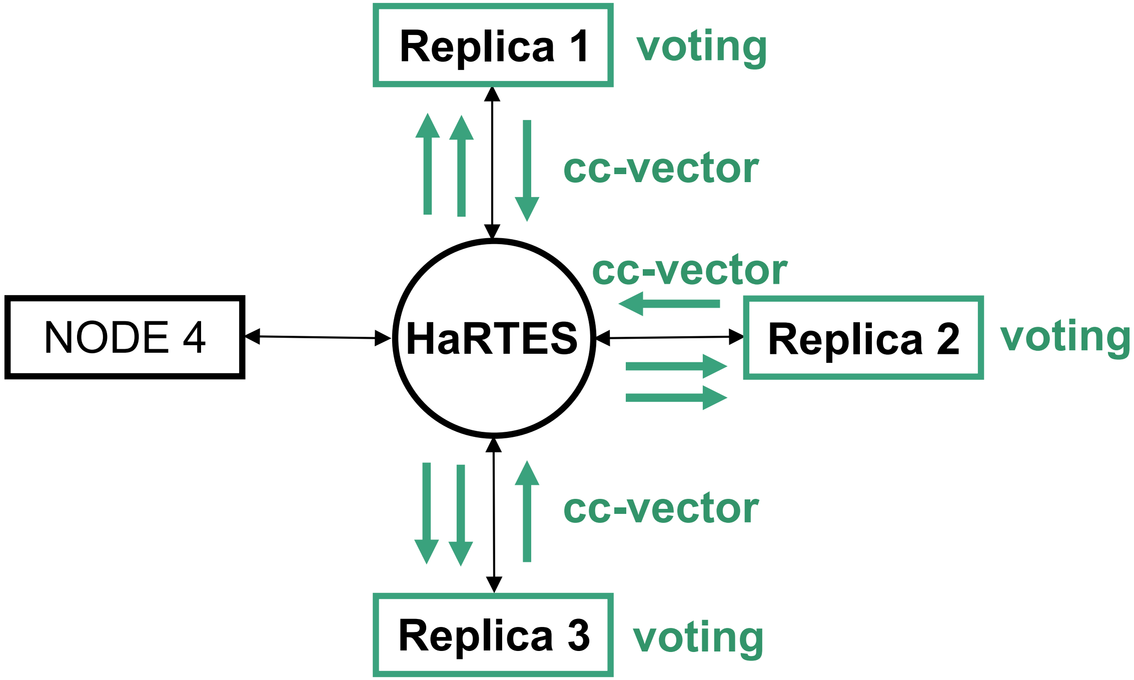
Node (FTT slave) Replication

- **active** replication for the slaves
  - using **N-Version Programming** terminology



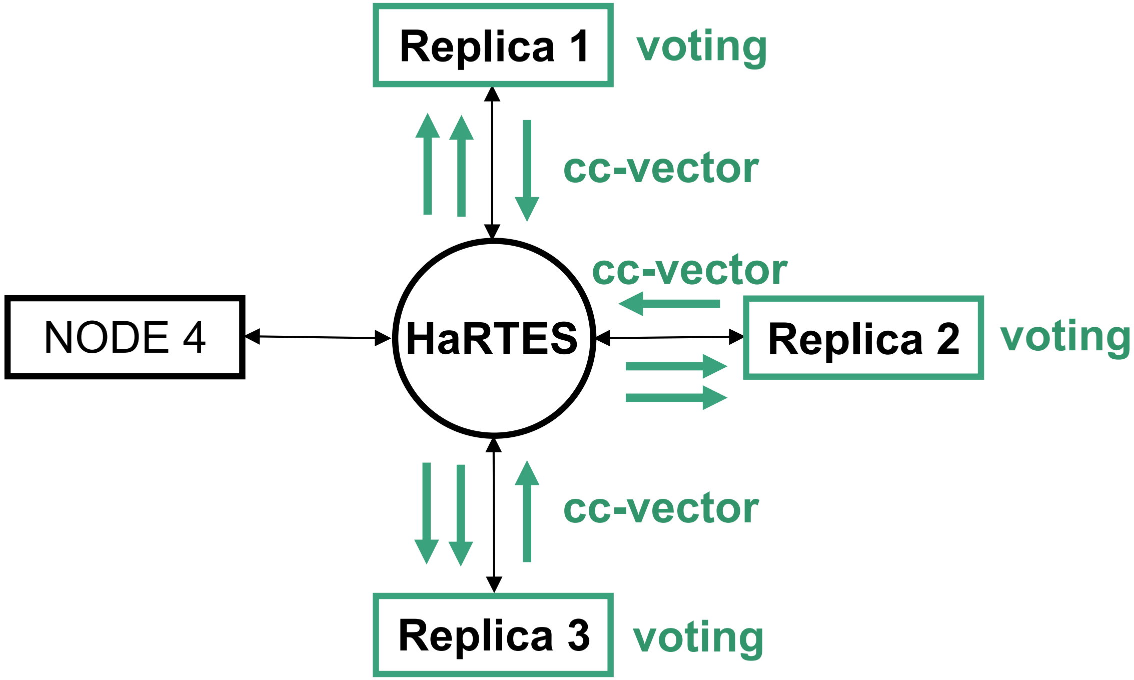
Node (FTT slave) Replication

- **active** replication for the slaves. **Main Issues**:
  - **Synchronization** among replicated tasks ("CAMBADA-style")
  - **Independence of failures** has to be ensured among replicas
  - Voting has to be **consistent** (the replica determinism problem)



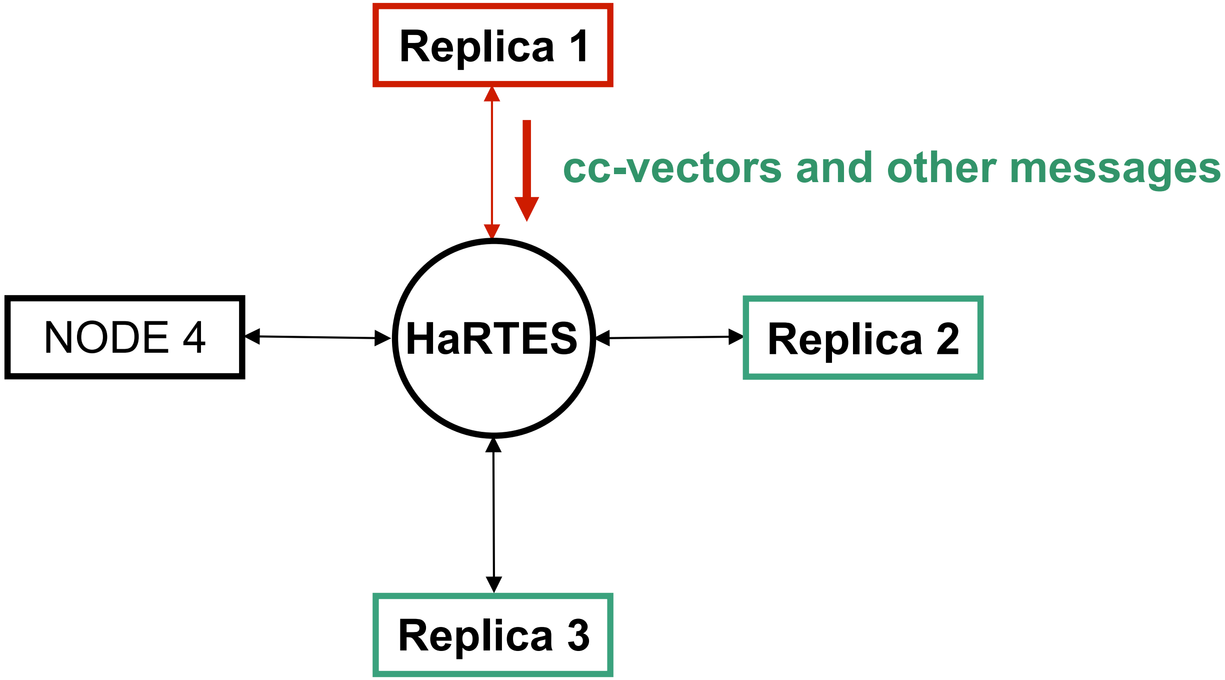
Node (FTT slave) Replication

- **active** replication for the slaves. **Main Issues**:
  - **Synchronization** among replicated tasks ("CAMBADA-style")
  - **Independence of failures** has to be ensured among replicas
  - Voting has to be **consistent** (the replica determinism problem)



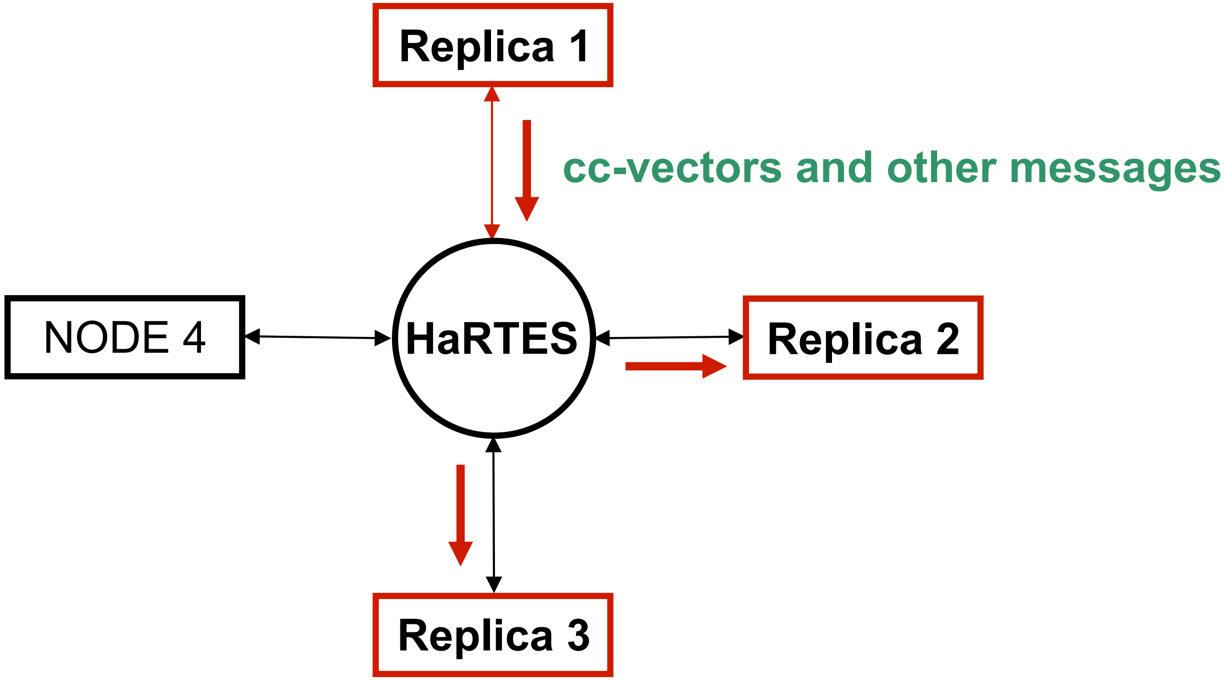
Independence of Failures

- **Two aspects**:
  - Replicas in different nodes, thus initially they are independent
  - However, a faulty replica or link can propagate errors



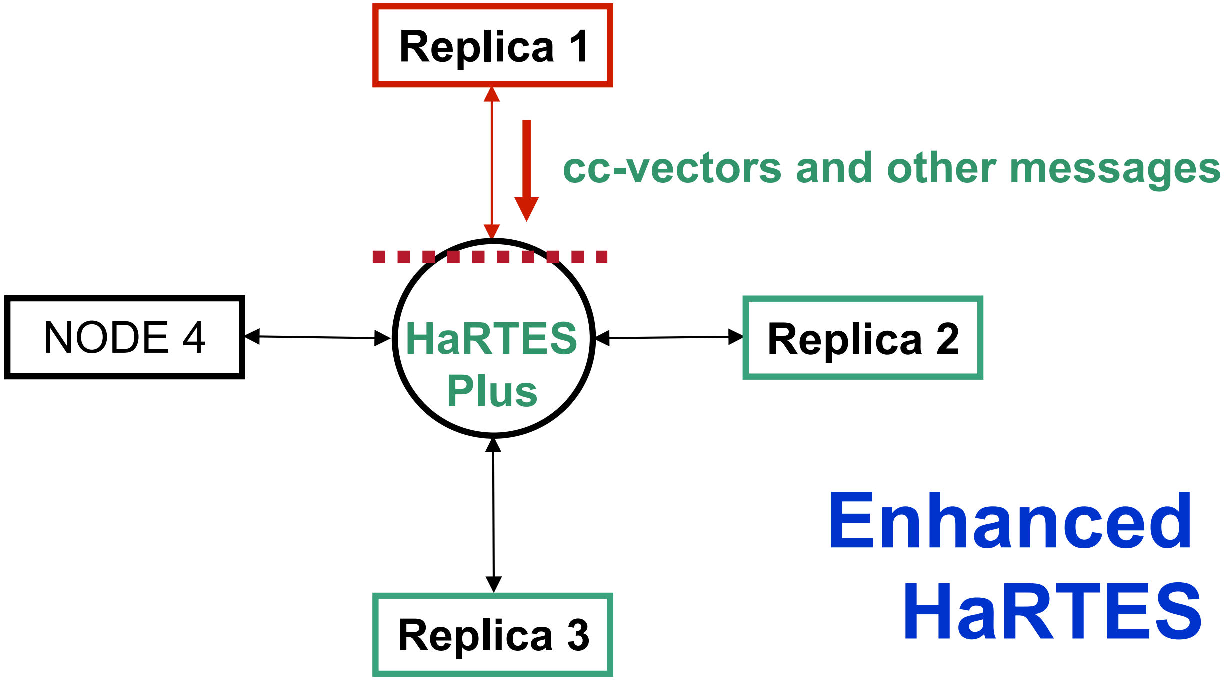
Independence of Failures

- **Two aspects**:
  - Replicas in different nodes, thus initially they are independent
  - However, a faulty replica or link can propagate errors



Independence of Failures

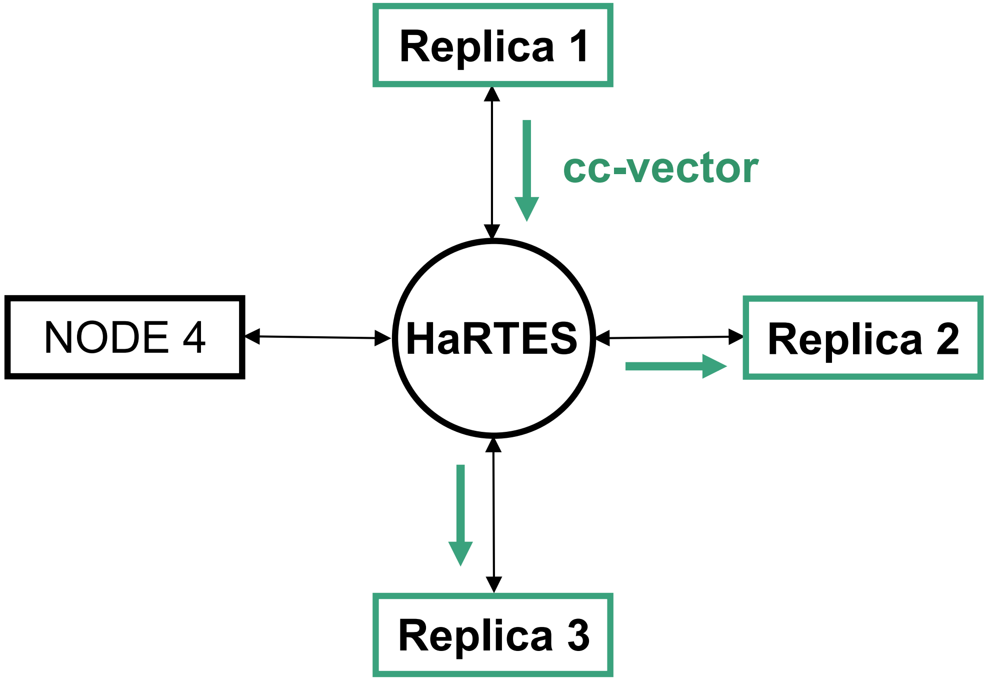
- **Two aspects**:
  - Replicas in different nodes, thus initially they are independent
  - However, a faulty replica or link can propagate errors. **Prevent it!!**



Enhanced HaRTES switch (port guardians)

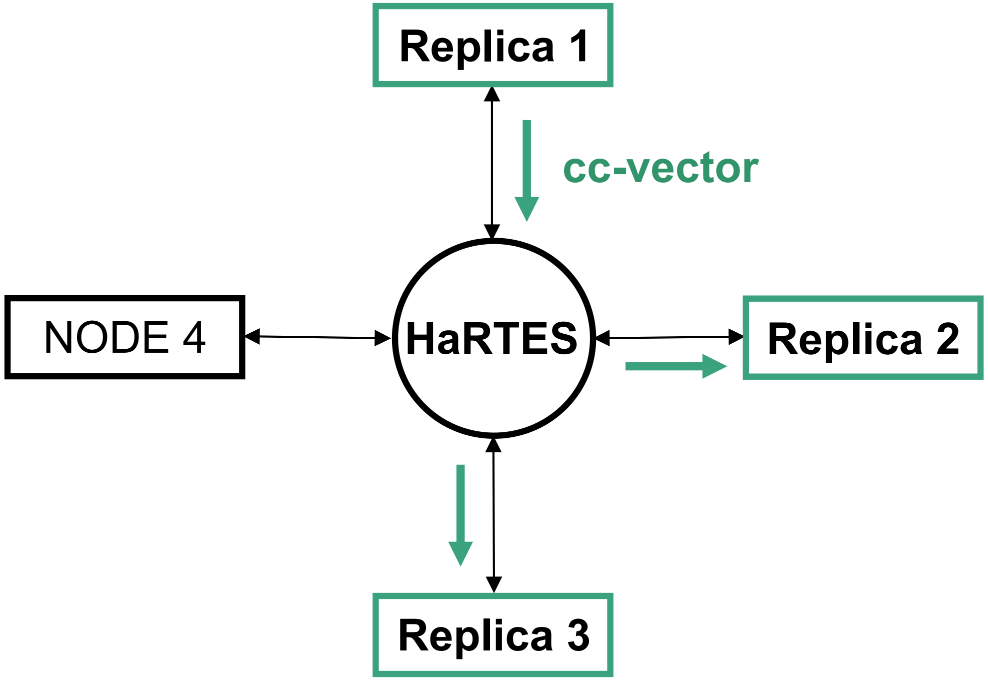
Consistent Voting

- Receiving the same cc-vectors helps!



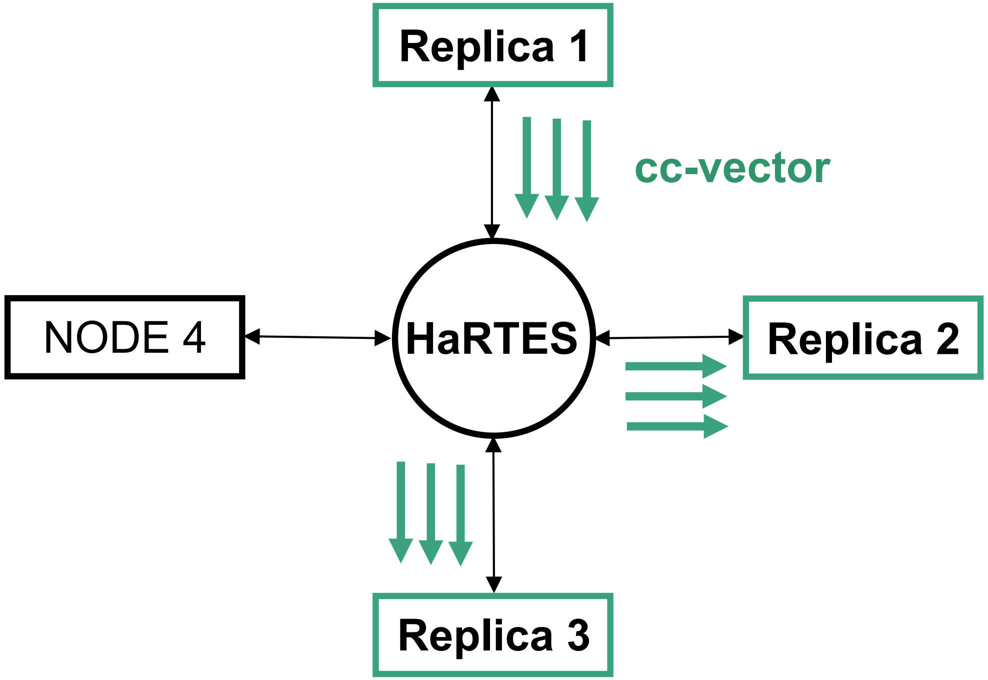
Consistent Voting

- **2nd choice:** Use **proactive retransmissions** & topology
  - Send each cc-vector enough times to tolerate transient faults in links
  - If Replica 1 is faulty (trans. or perm.) it reaches or not HaRTES
  - If it reaches HaRTES, this one sends enough times to R2 and R3



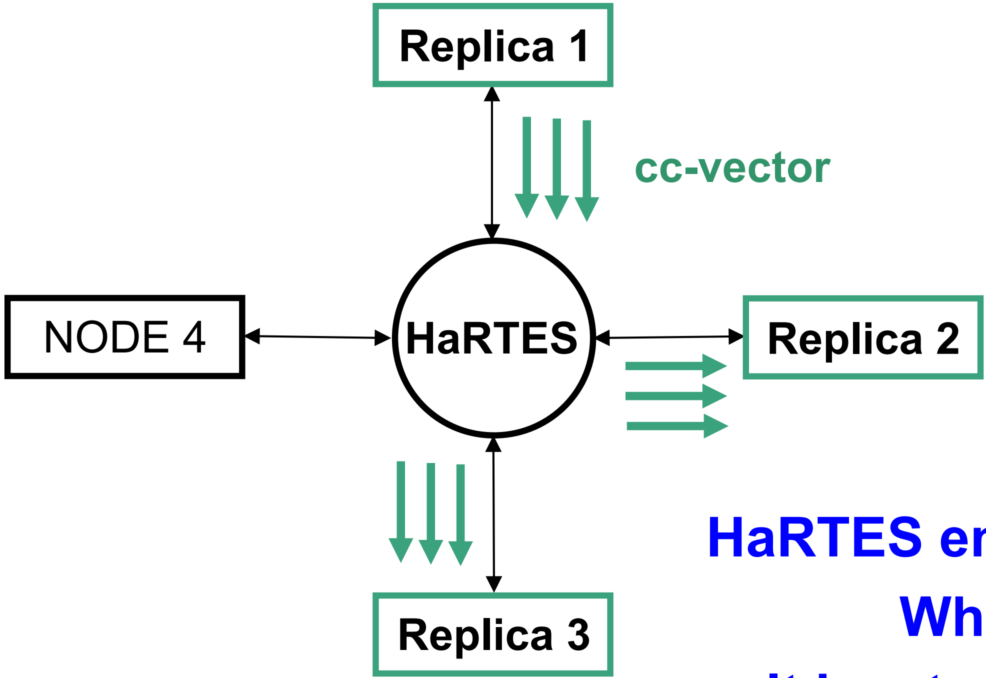
Consistent Voting

- **2nd choice:** Use **proactive retransmissions** & topology
  - Send each cc-vector enough times to tolerate transient faults in links
  - If Replica 1 is faulty (trans. or perm.) it reaches or not HaRTES
  - If it reaches HaRTES, this one sends enough times to R2 and R3



Consistent Voting

- **2nd choice:** Use **proactive retransmissions** & topology
  - Send each cc-vector enough times to tolerate transient faults in links
  - If Replica 1 is faulty (trans. or perm.) it reaches or not HaRTES
  - If it reaches HaRTES, this one sends enough times to R2 and R3



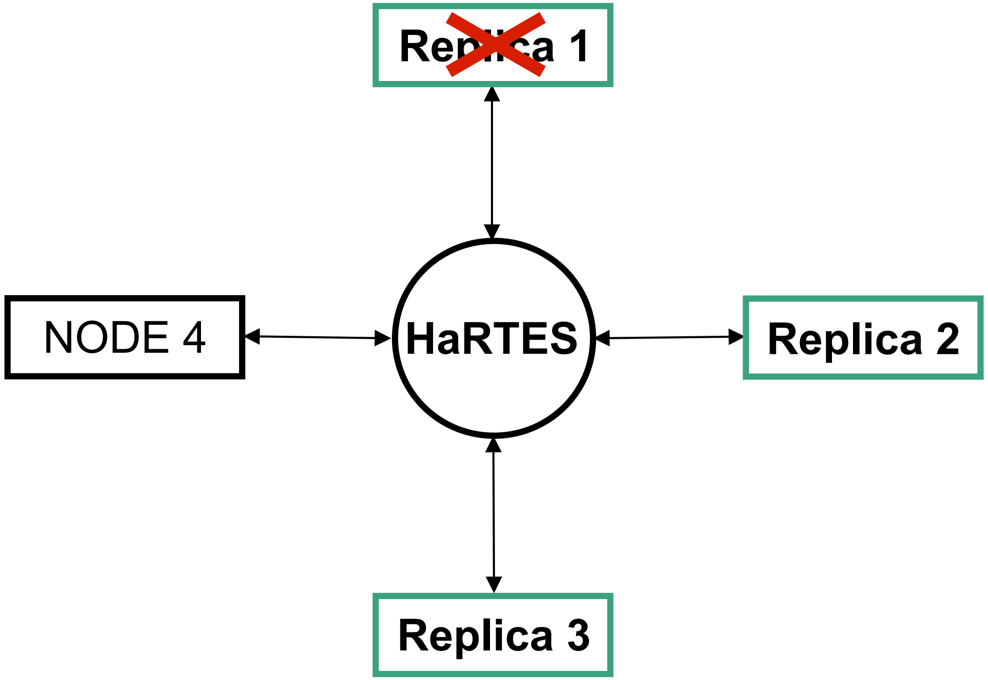
HaRTES ensures consistency

What if it fails?

It has to be fault tolerant!

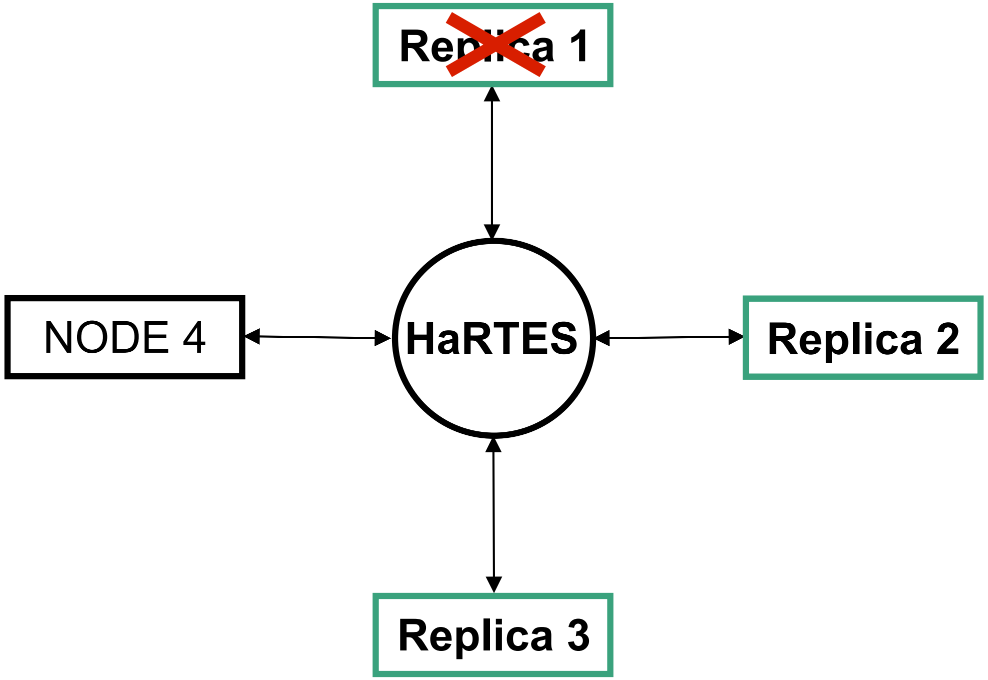
Node (FTT slave) Reintegration

- **Using active** replication for the slaves...
  - When a replica is permanently faulty, we lose fault tolerance
  - **We do not want the same when the fault is not permanent**



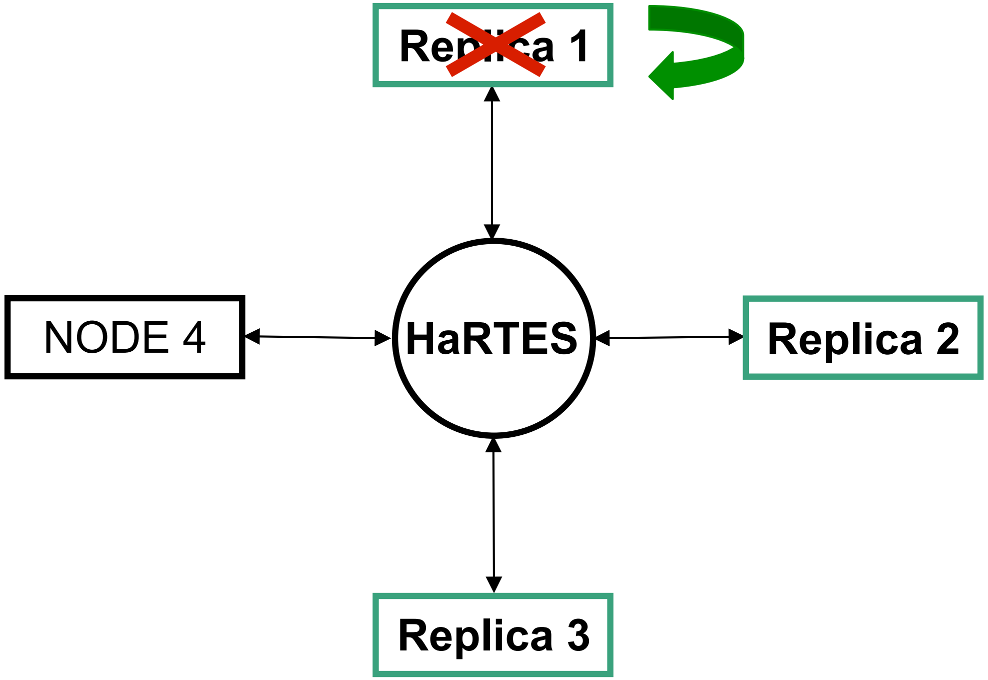
Node (FTT slave) Reintegration

- **1st: Try to reset the node**
  - When node counters reach a threshold node should auto-reset
  - When HaRTES counters reach a threshold a reset cmd. is sent
  - A watchdog timer is attached to each node and resets if no TM



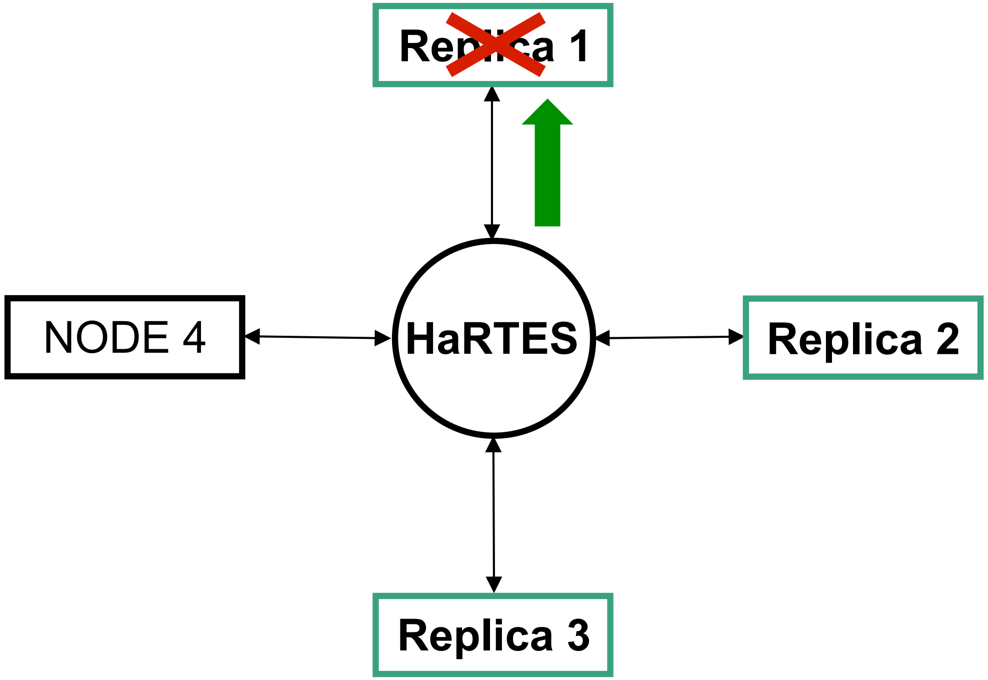
Node (FTT slave) Reintegration

- **1st: Try to reset the node**
  - When node counters reach a threshold node should auto-reset
  - When HaRTES counters reach a threshold a reset cmd. is sent
  - A watchdog timer is attached to each node and resets if no TM



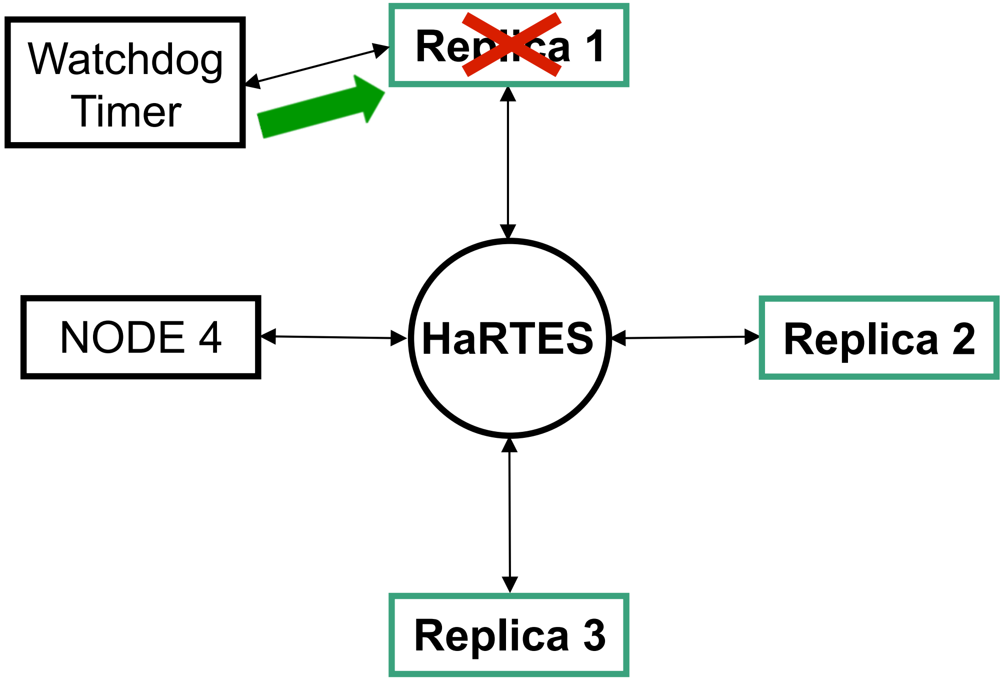
Node (FTT slave) Reintegration

- **1st: Try to reset the node**
  - When node counters reach a threshold node should auto-reset
  - When HaRTES counters reach a threshold a reset cmd. is sent
  - A watchdog timer is attached to each node and resets if no TM



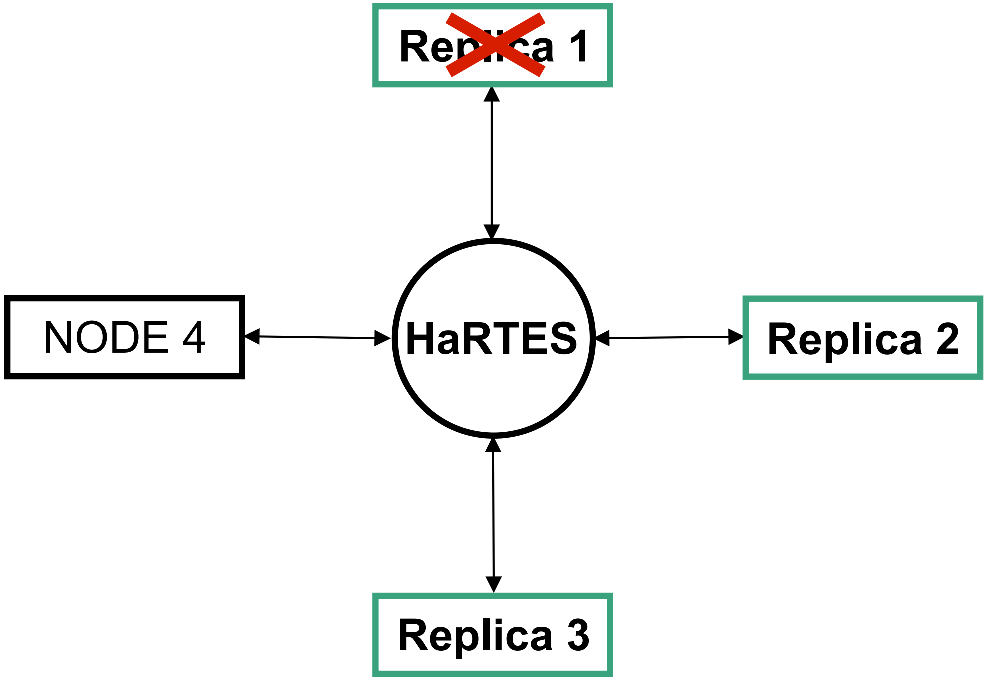
Node (FTT slave) Reintegration

- **1st: Try to reset the node**
  - When node counters reach a threshold node should auto-reset
  - When HaRTES counters reach a threshold a reset cmd. is sent
  - A watchdog timer is attached to each node and resets if no TM



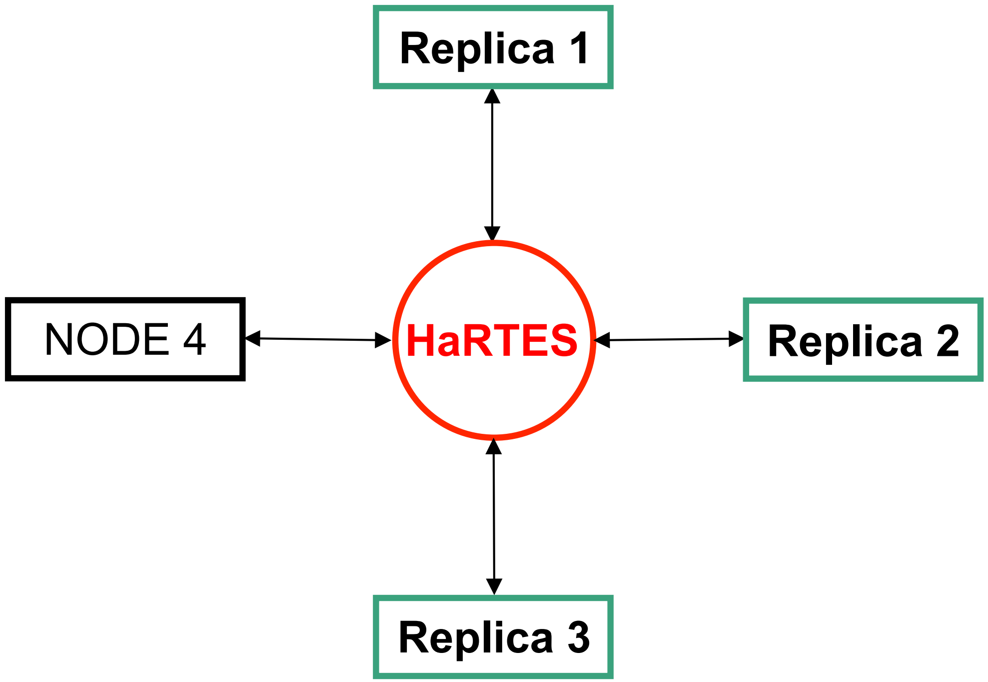
Node (FTT slave) Reintegration

- **2nd: The node gets the reintegration information**
  - We only consider simple control tasks such as a PID
  - All (small) info on the state of the task is exchanged in each cc-vec
  - The reset node receives the info from the others, votes & uses it



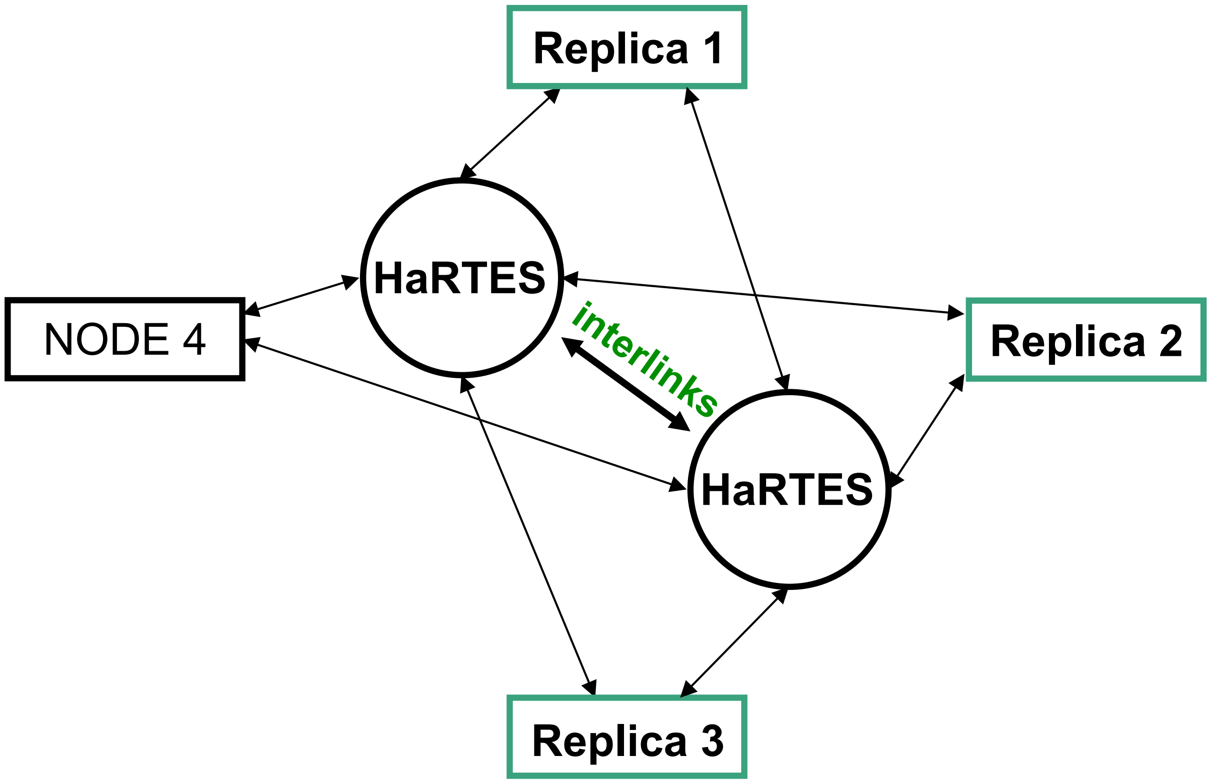
Channel Replication

- Otherwise, **master and switch** are **single points of failure**



Channel (Spatial) Replication

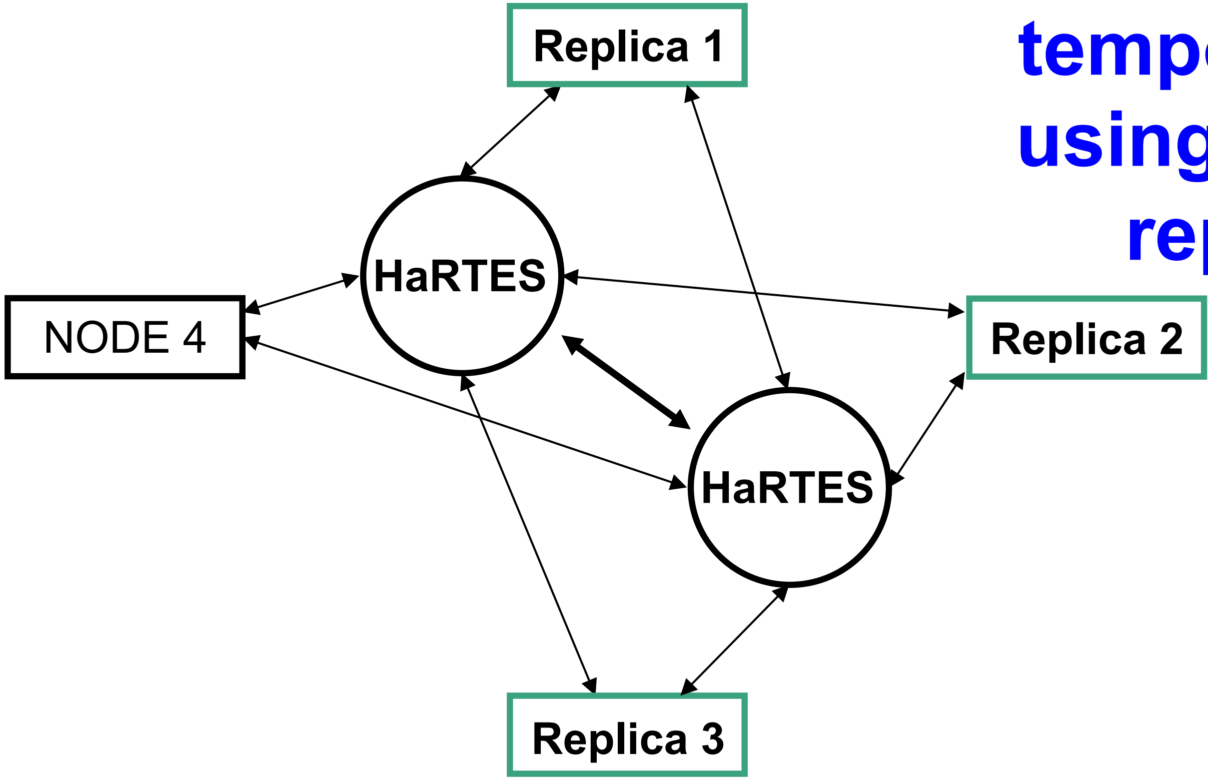
- Otherwise, **master and switch** are **single points of failure**
  - Therefore we replicate as shown below



Channel (Spatial) Replication

- Otherwise, **master and switch** are **single points of failure**
  - Therefore we replicate as shown below

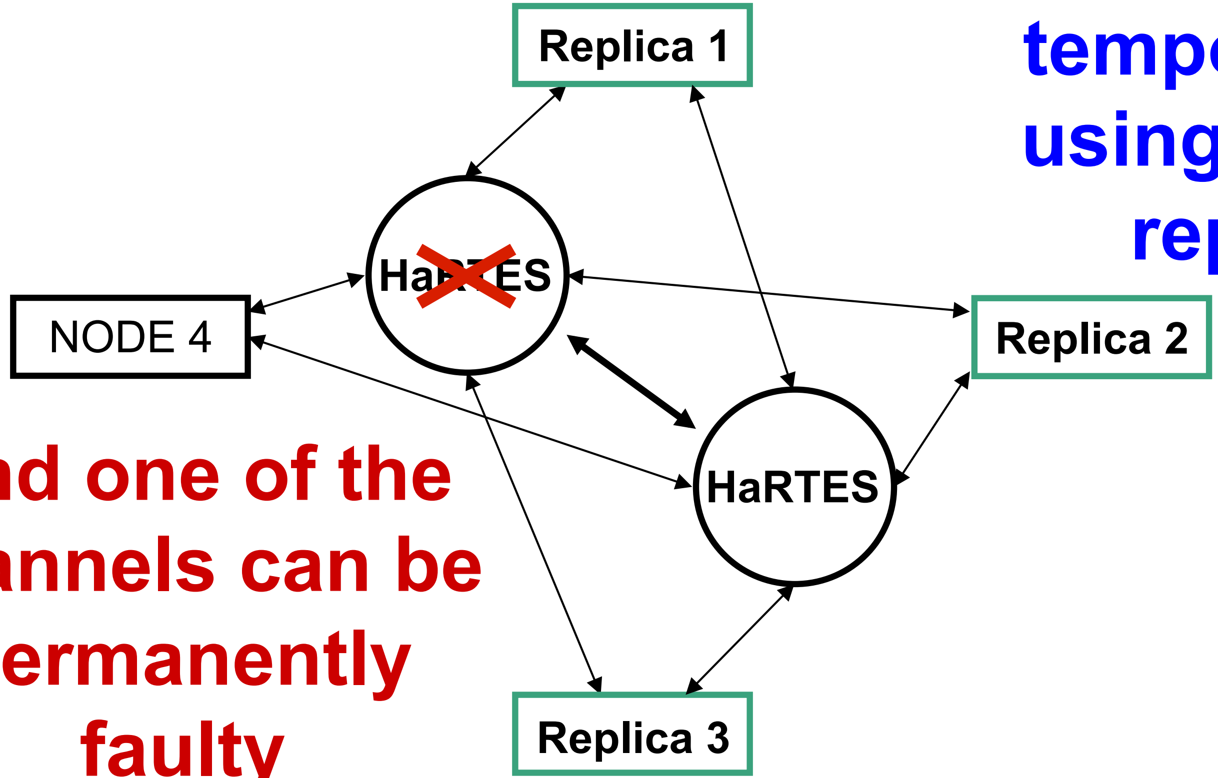
BUT it is a waste to tolerate temporary faults using the spatial replication



Channel (Spatial) Replication

- Otherwise, **master and switch** are **single points of failure**
  - Therefore we replicate as shown below

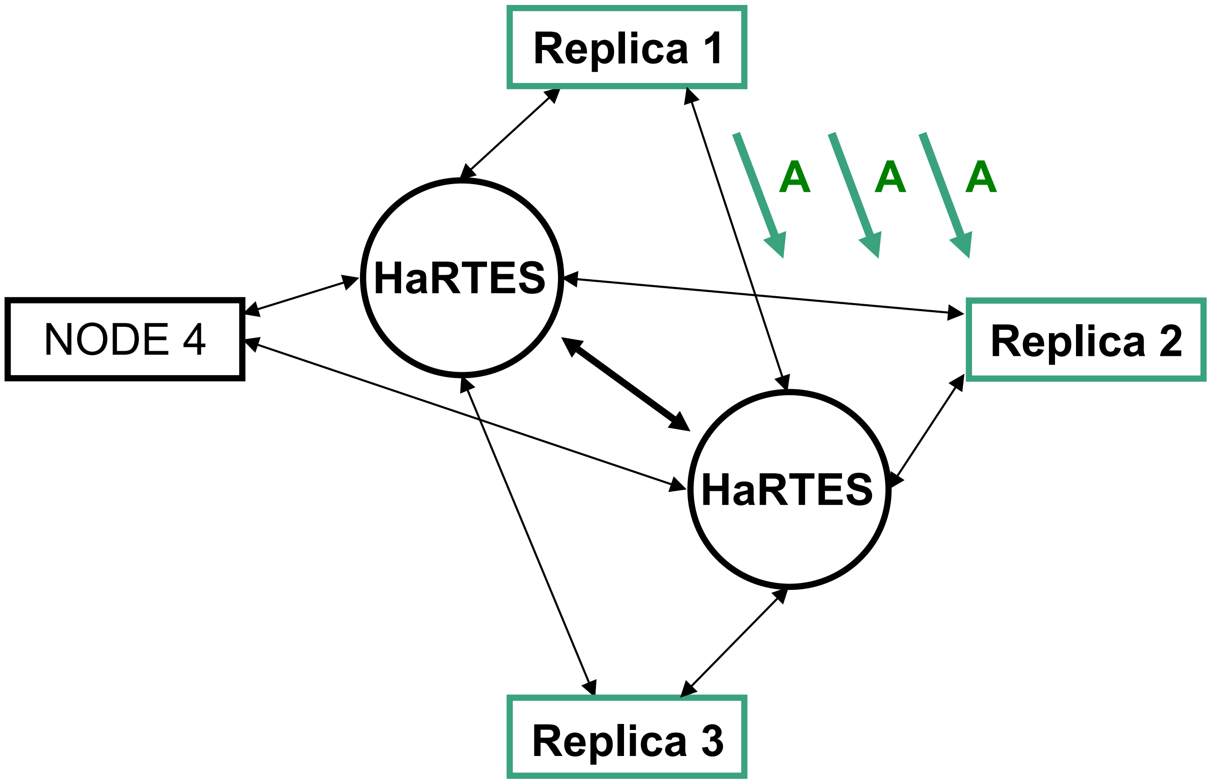
BUT it is a waste to tolerate temporary faults using the spatial replication



And one of the channels can be permanently faulty

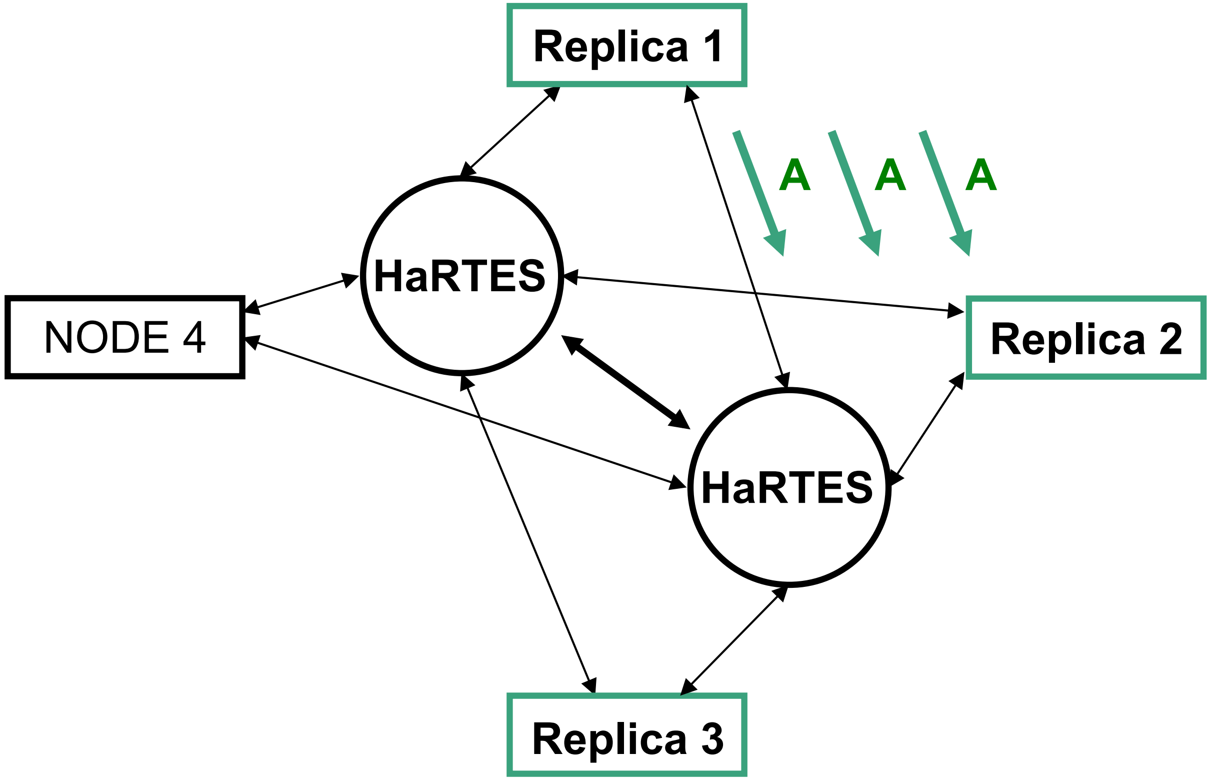
Channel (Temporal) Replication

- Therefore we will use **temporal redundancy for msgs**
  - We chose **proactive retransmissions** for its easier schedulability



Channel (Temporal) Replication

- Therefore we will use **temporal redundancy for msgs**
  - We chose **proactive retransmissions** for its easier schedulability
  - **Predictable and deterministic**



Channel (Temporal) Replication

- More specifically for **slaves' regular messages**
  - Addition of **message redundancy level** in the spec

$$\mathrm{SRT} = \{m_i \mid m_i = (C_i, D_i, T_i, O_i, P_i), i \in [1, N_\mathrm{S}]\}$$

$$\mathrm{ART} = \{m_i \mid m_i = (C_i, D_i, I_i, P_i), i \in [1, N_\mathrm{A}]\}.$$

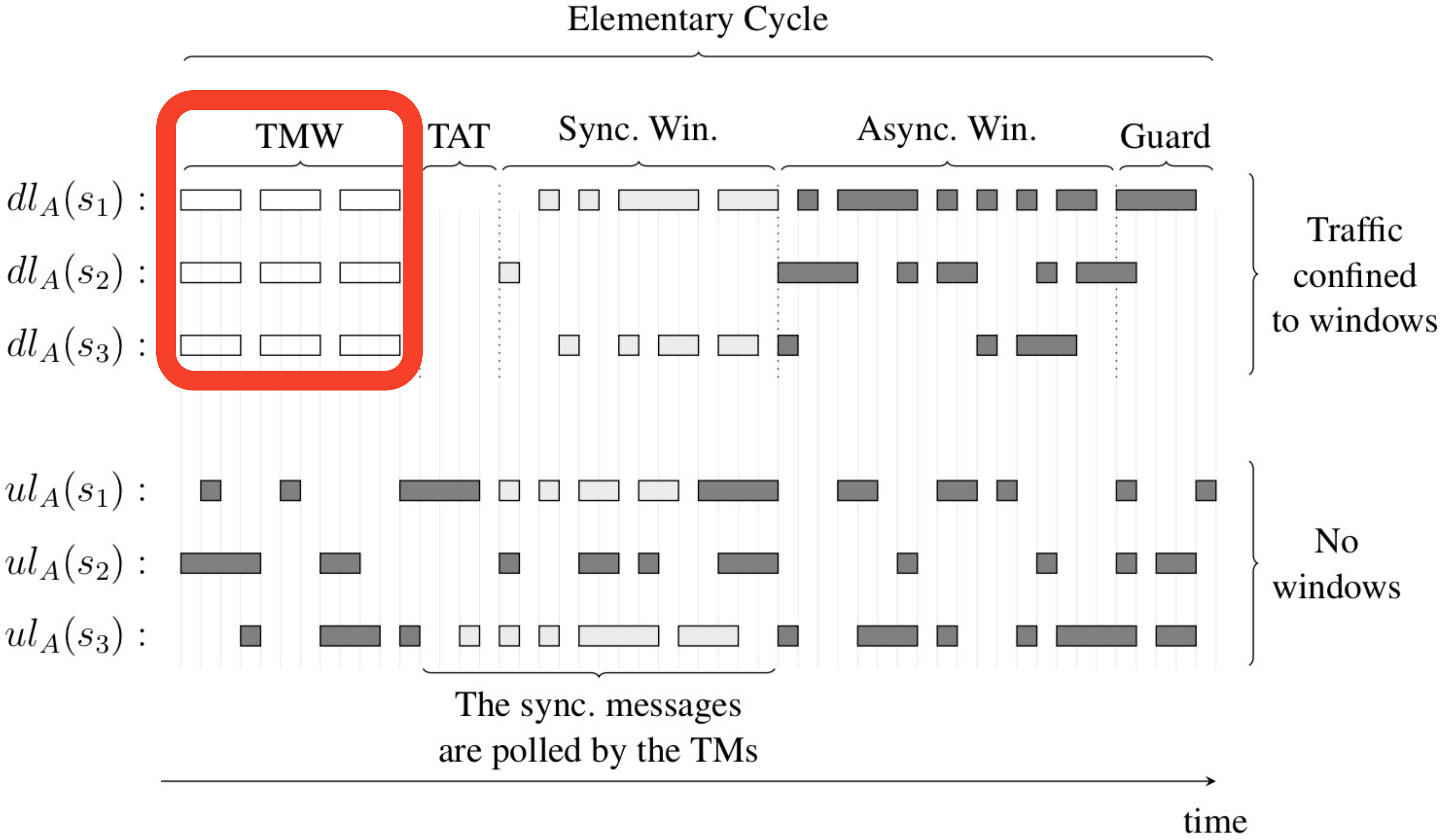
$$\Downarrow$$

$$\mathrm{SRT} = \{m_i \mid m_i = (C_i, D_i, T_i, O_i, P_i, k_i), i \in [1, N_\mathrm{S}]\}$$

$$\mathrm{ART} = \{m_i \mid m_i = (C_i, D_i, I_i, P_i, k_i), i \in [1, N_\mathrm{A}]\}.$$

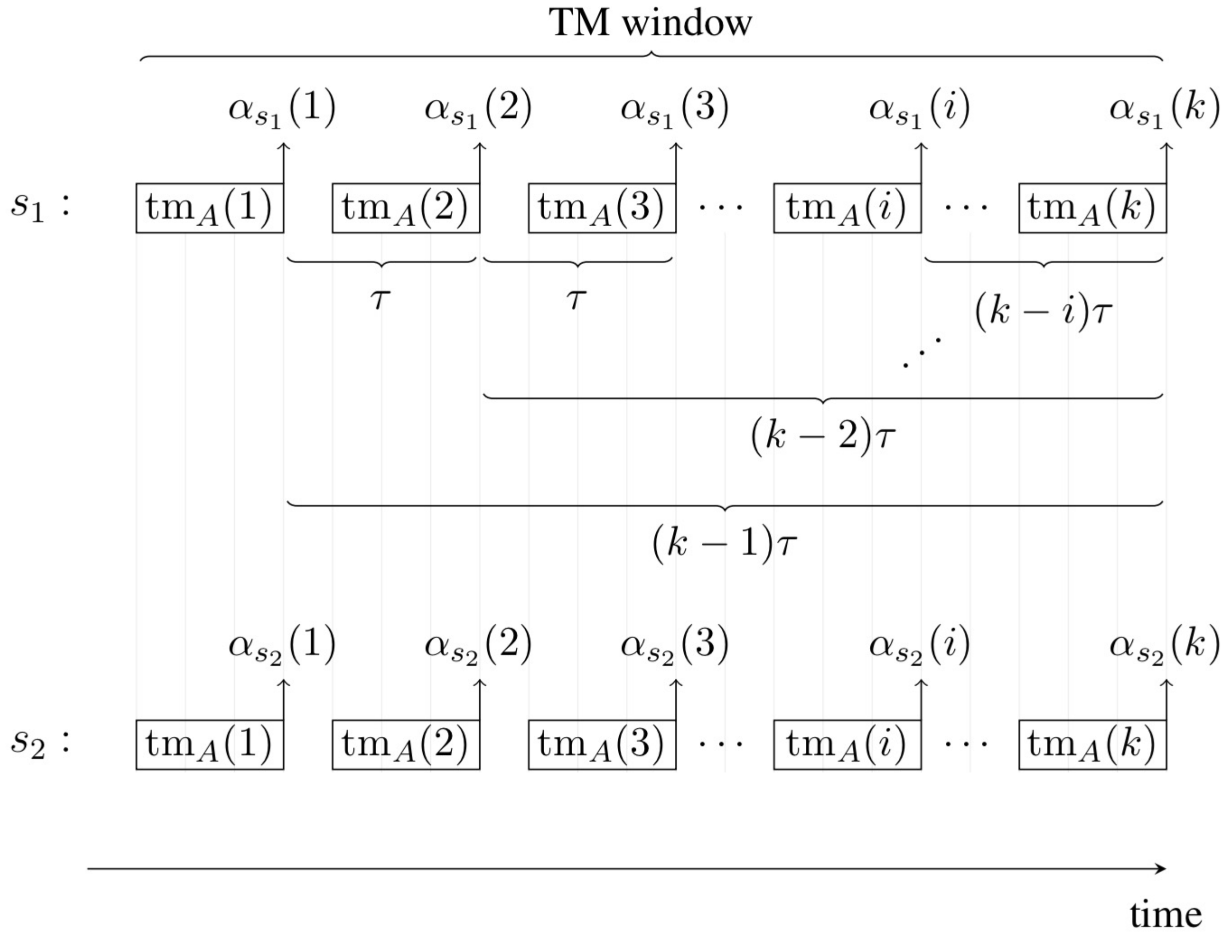
Channel (Temporal) Replication

- More specifically for **masters' trigger message**
  - Multiple TMs per Trigger Message Window



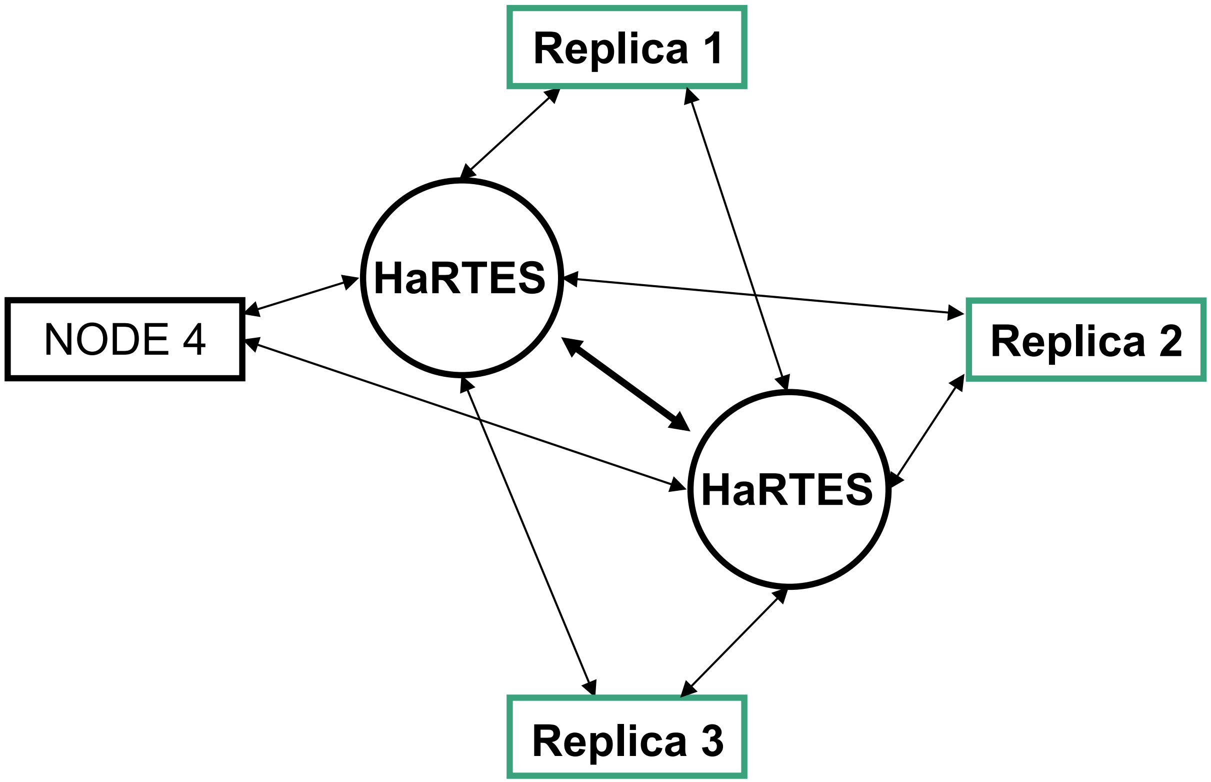
Channel (Temporal) Replication

- Therefore this changes the **EC synchronization w. slaves:**
  - Isochronous TM transmission; receiving any replica there is sych



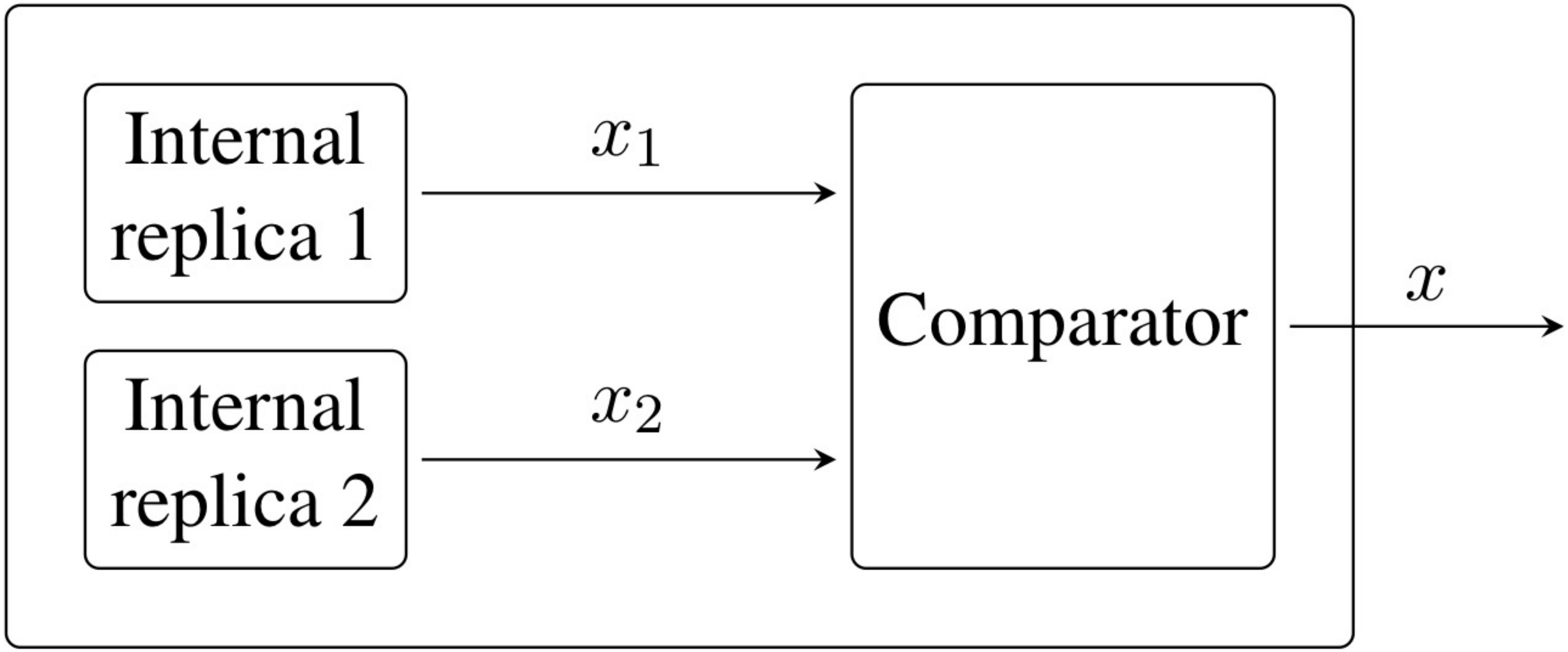
Channel (Spatial) Replication

- Back to spatial replication it is necessary to note that by **replicating the switch we also replicated the links**
  - On the one hand, we have **tolerance to faults in links**
  - On the other hand, there are **more chances for error propagation**



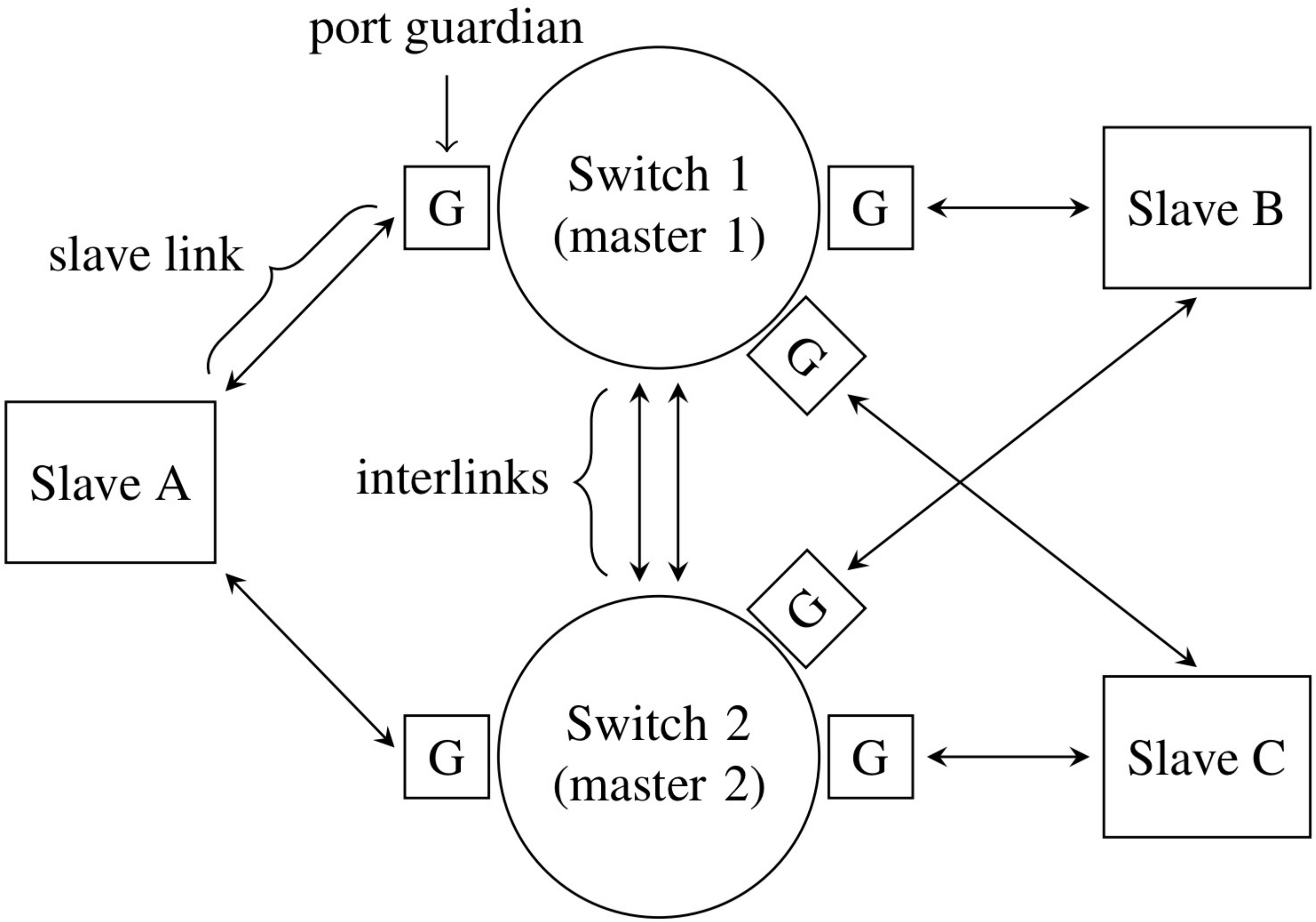
Channel (Spatial) Replication

- **Restriction of switch failure semantics:** internal duplication & comparison
  - Not implemented



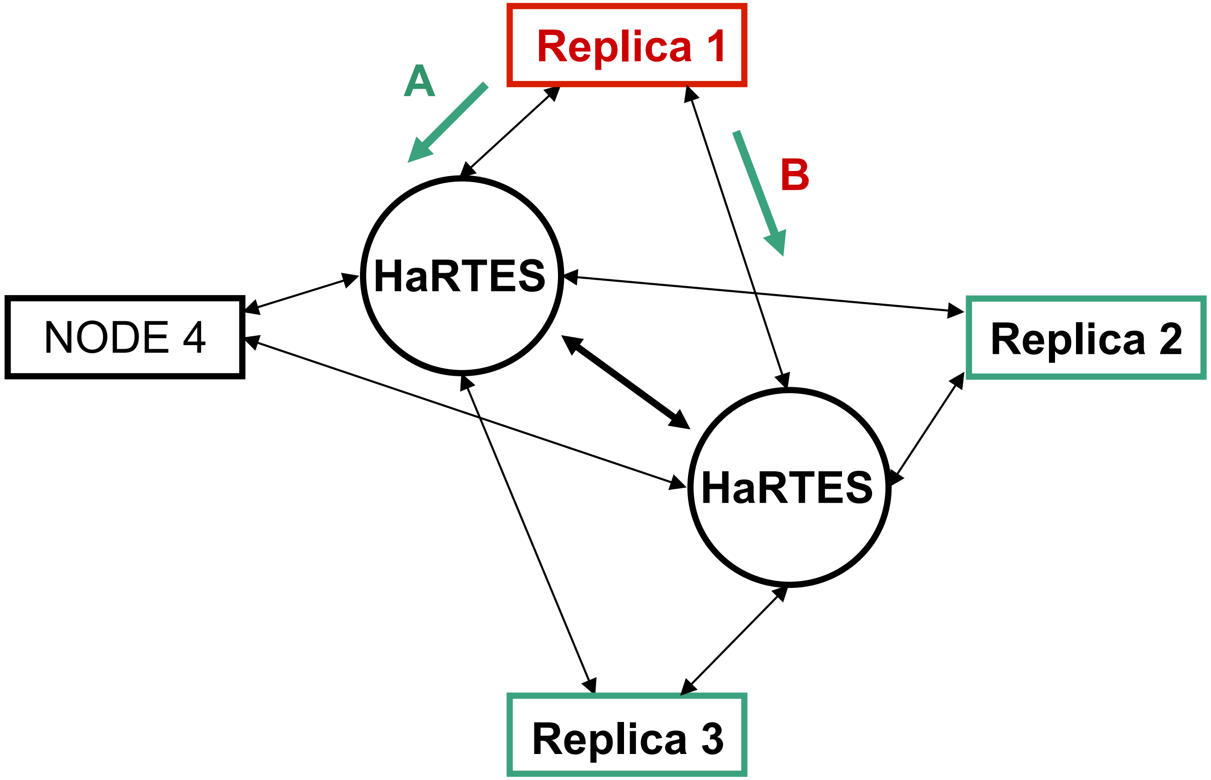
Channel (Spatial) Replication

- **Restriction of <u>slave</u> failure semantics: port guardians**



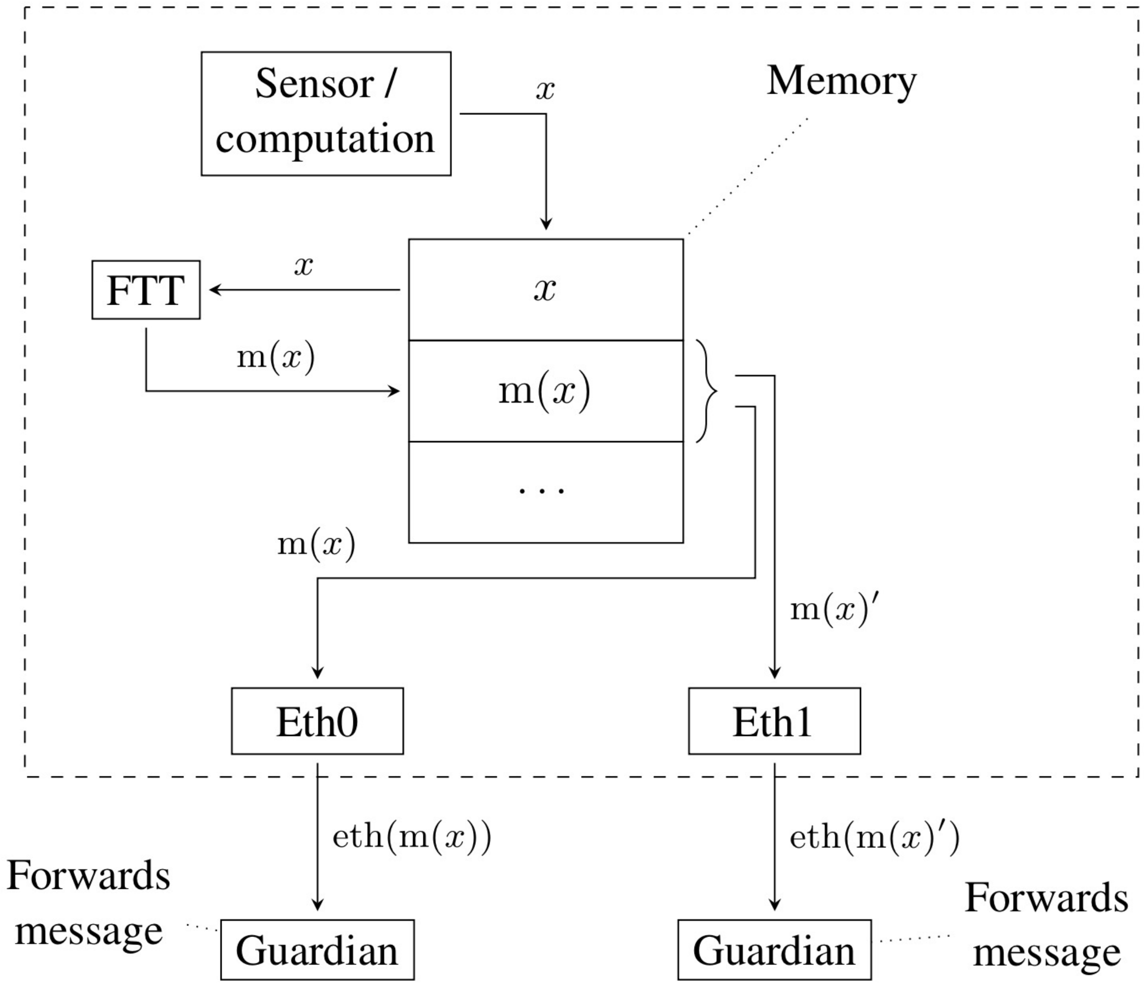
Channel (Spatial) Replication

- **Restriction of <u>slave</u> f-semantics: elimination of 2-faced**
  - A **faulty node** (transient of permanent) could send different replicas of a msg to each HaRTES.
  - **Undetectable with FTT rules!**



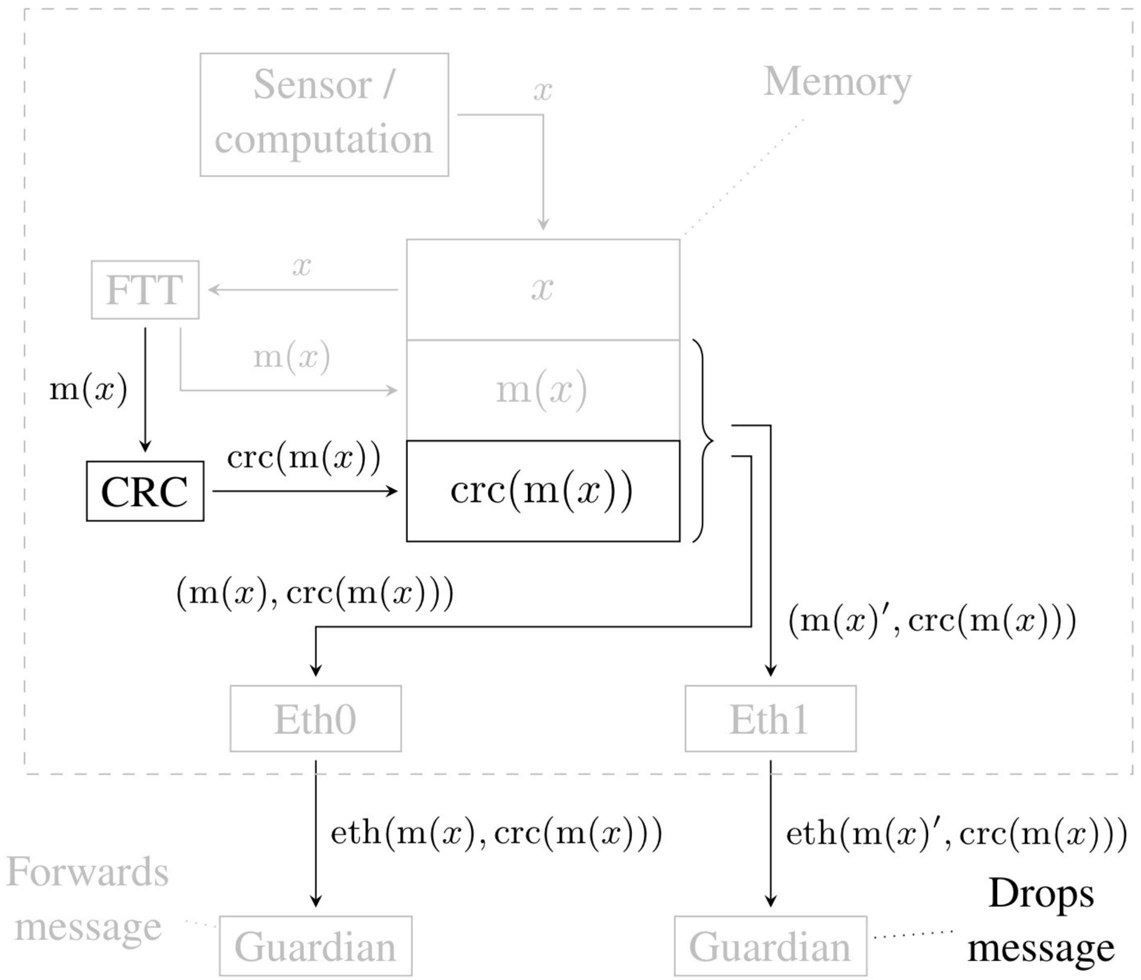
Channel (Spatial) Replication

- **Restriction of slave f-semantics: elimination of 2-faced**



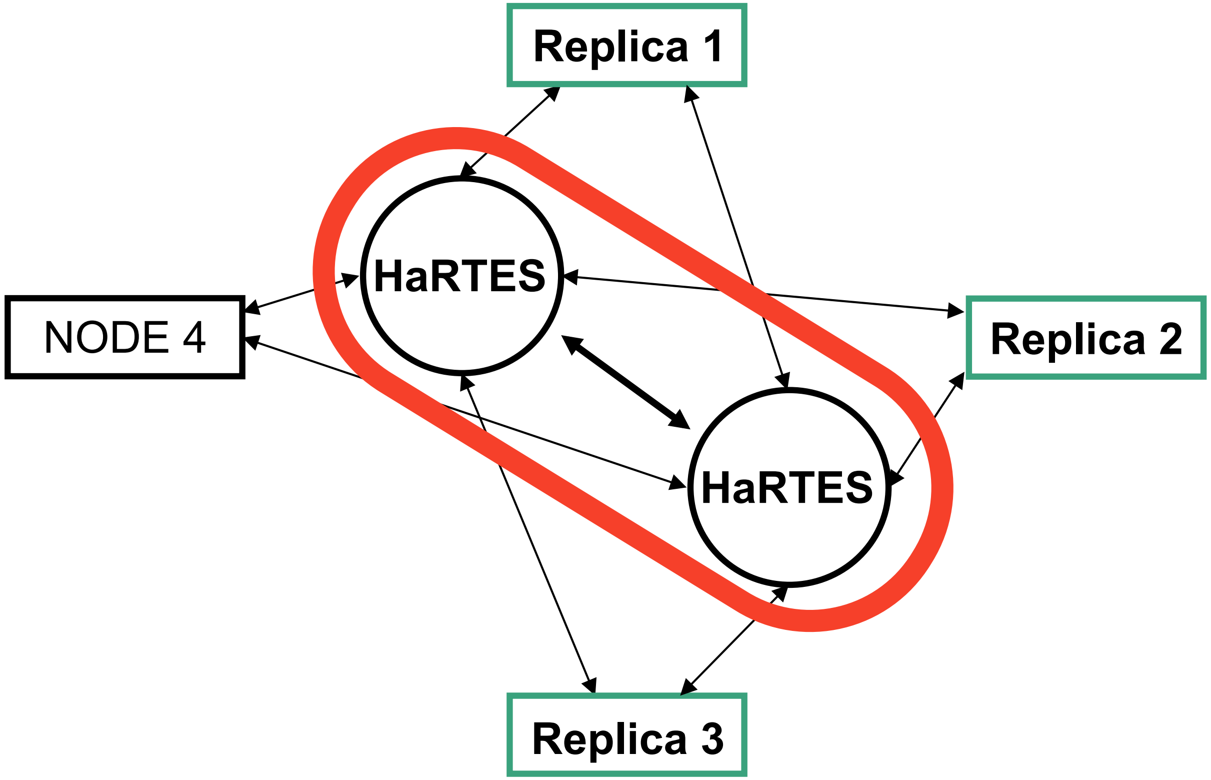
Channel (Spatial) Replication

- **Restriction of <u>slave</u> f-semantics: elimination of 2-faced**



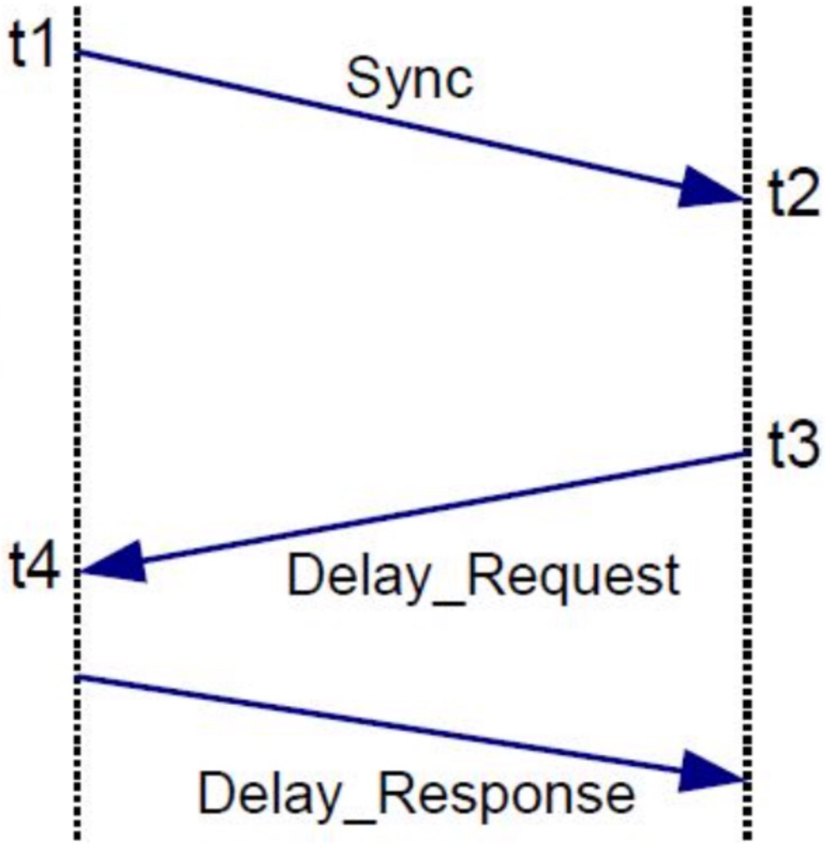
Channel (Spatial) Replication

- Additionally, spatial replication calls for **managing the replication** of the different components
  - Issues related to **replica determinate the master replicas**



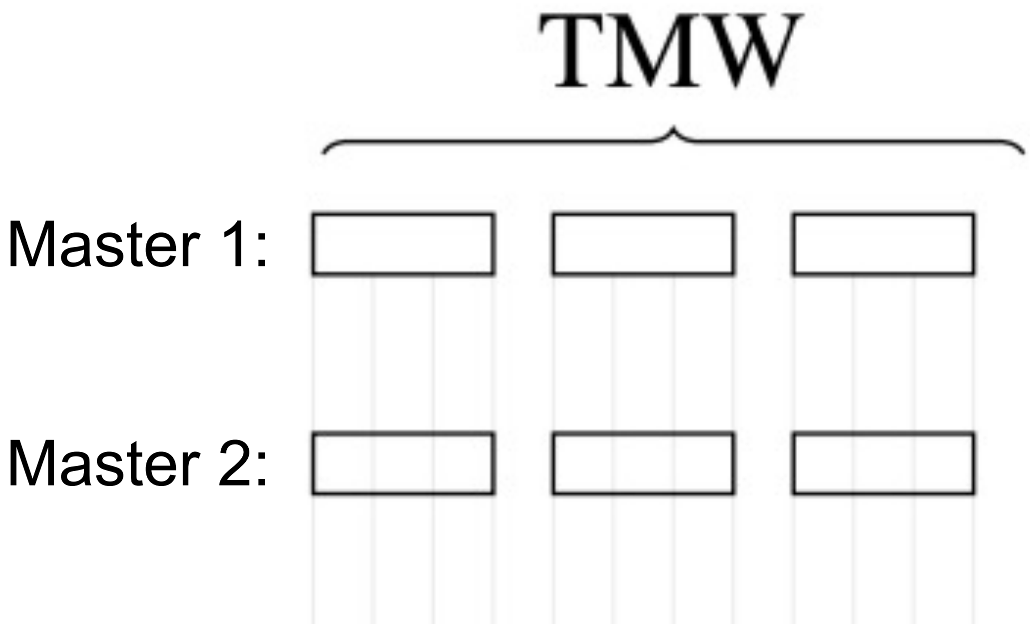
Channel (Spatial) Replication

- **Master replica determinism (time domain):**
  - Leader-follower approach with a rendez-vous based on PTP



Channel (Spatial) Replication

- **Master replica determinism (time domain):**
  - Lock-step transmission of TMs



Channel (Spatial) Replication

- **Master replica determinism (<u>value</u> domain):**
  - Initial conditions

Start with consistent SRDBs + no internal non-determinism

If t = 0, then

SRDB of master 1 = SRDB of master 2

Channel (Spatial) Replication

- **Master replica determinism (value domain):**
  - Ensure **consistent updates** of **SRDBs (in the masters)**

If t > 0 and both masters not faulty, then

SRDB of master 1 updated iff
SRDB of master 2 is also updated.
(Reliably exchange min pending update request on interlinks)

Channel (Spatial) Replication

- **Master replica determinism (value domain):**
  - Synchronized and consistent **NRDB** updates **(in the slaves)**

Piggyback admission control results and NRDB update commands on reliable and synchronized TMs

Prototype

•Master + Switch

- Intel Core i7 → parallelize as much as possible
- 8 GB RAM
- Up to 18 NICs
  - 2 NICs Motherboard
  - Up to 16 NICs 4 x Intel I350 T4
- Ubuntu 12.04

•Main concerns

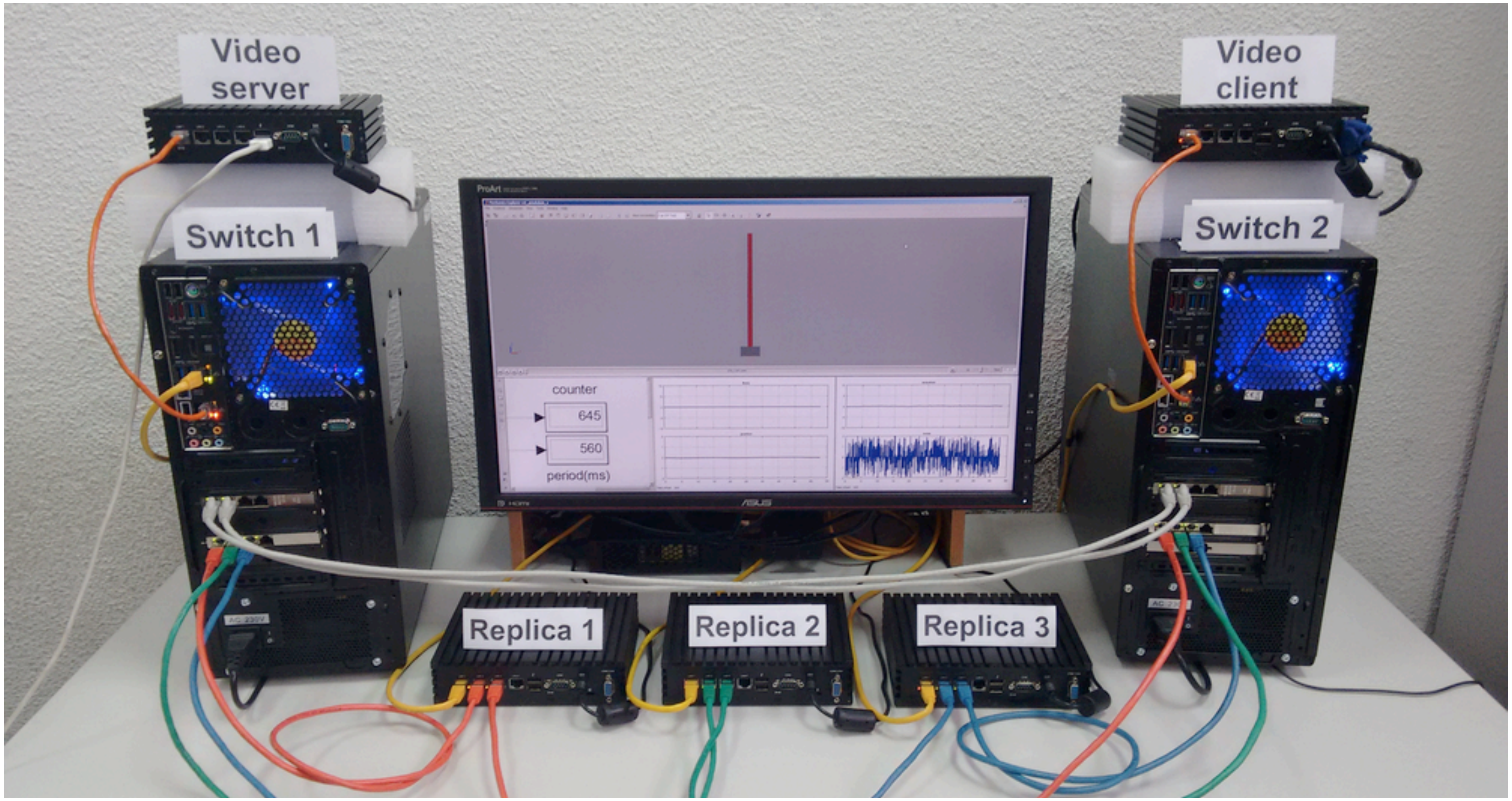
- OS determinism
  - Xenomai
- Network jitter
  - PF_RING → bypass network stack
  - Netw. teaming → Link repl. in kernel

•Slave

- Intel Atom D525
- 2 GB RAM
- 4 NICs
- Ubuntu 12.04

Demos

- https://www.youtube.com/watch?v=3THdUHuGMLI
- http://srv.uib.es/ft4ftt-final-prototype-demo/



Motivation of the **(current) DFT4FTT** project

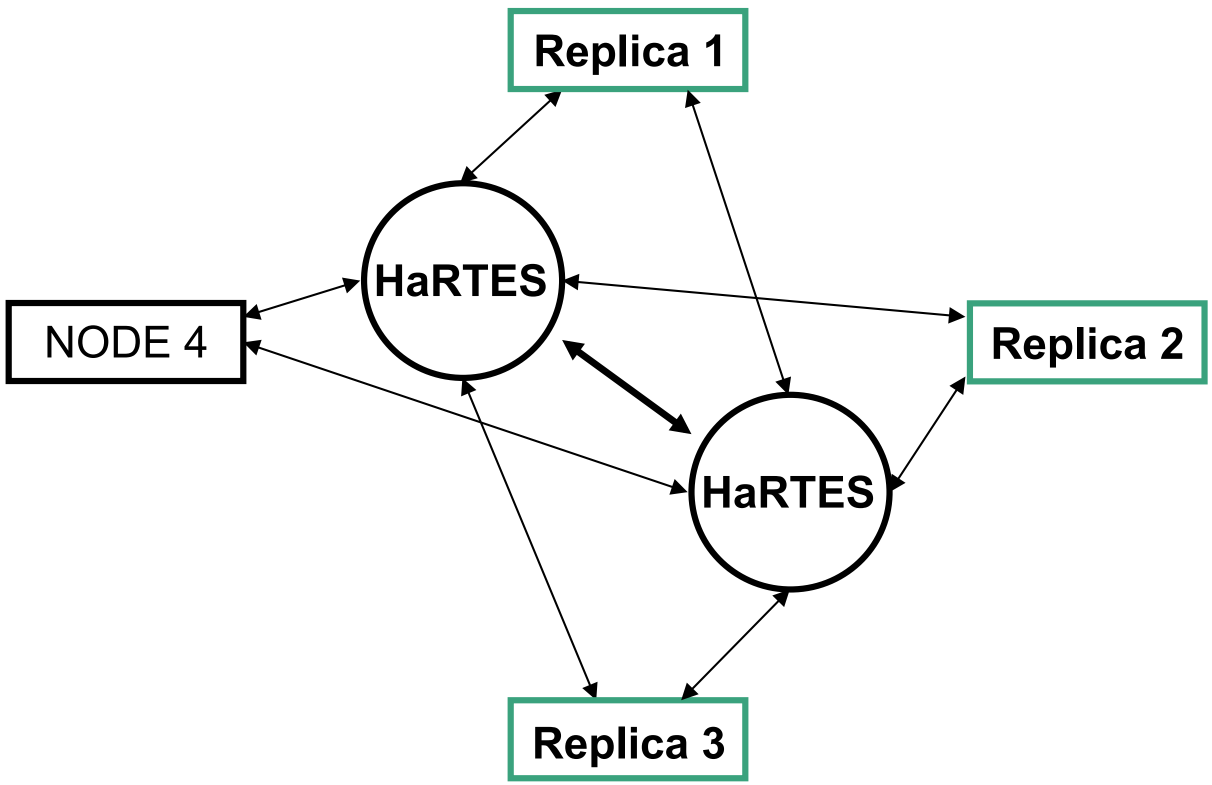
- There is always **a tradeoff** between flexibility and reliability
- However the **FT mechanisms** developed **in FT4FTT are static**, so flexibility can be improved by introducing **dynamic FT**
- Additionally **some parts of FTT** (the a-window) **were out of the scope** of FT4FTT!

Goal of the **DFT4FTT** project

- The design, implementation and validation of a highly-dependable communication infrastructure based on FTT-Ethernet **with a higher flexibility than FT4FTT.**
  - … by adding **Dynamic** Fault Tolerance Mechanisms

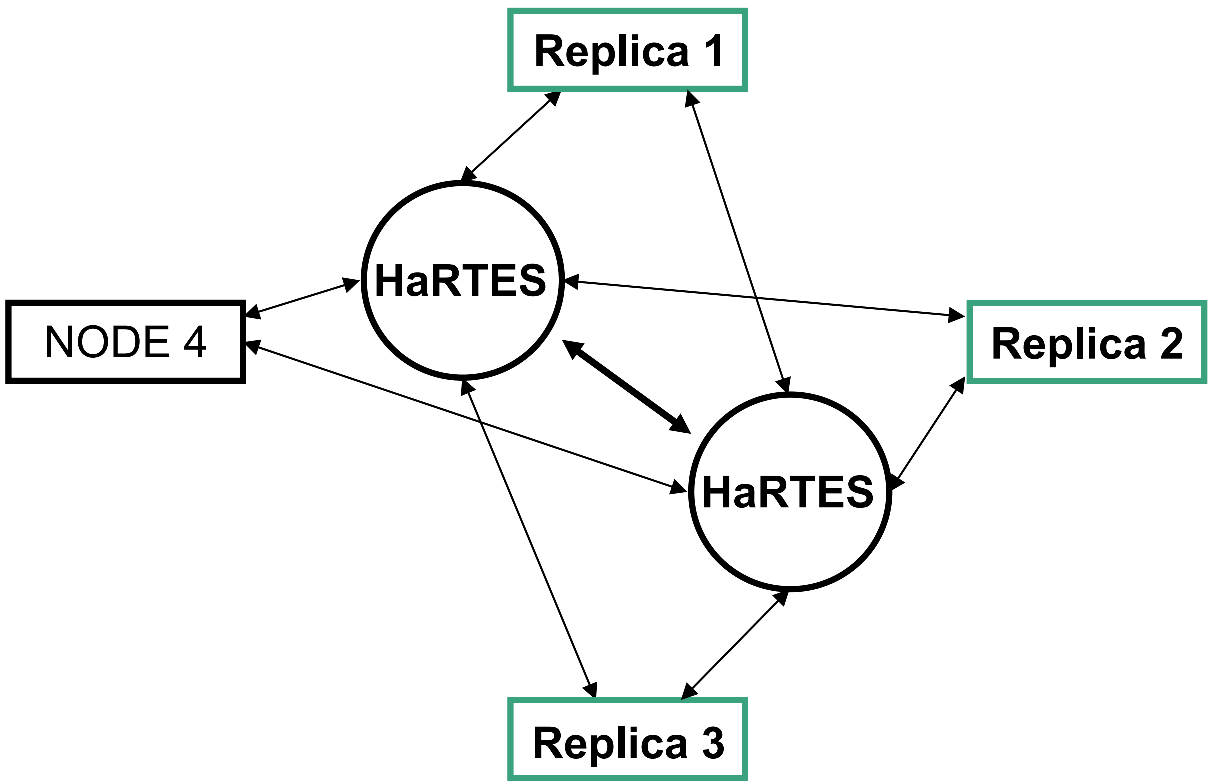
Dynamic Fault Tolerance

- **Targeted mechanisms**
  - Spatial replication of the nodes (tasks)
  - Temporal replication of the messages



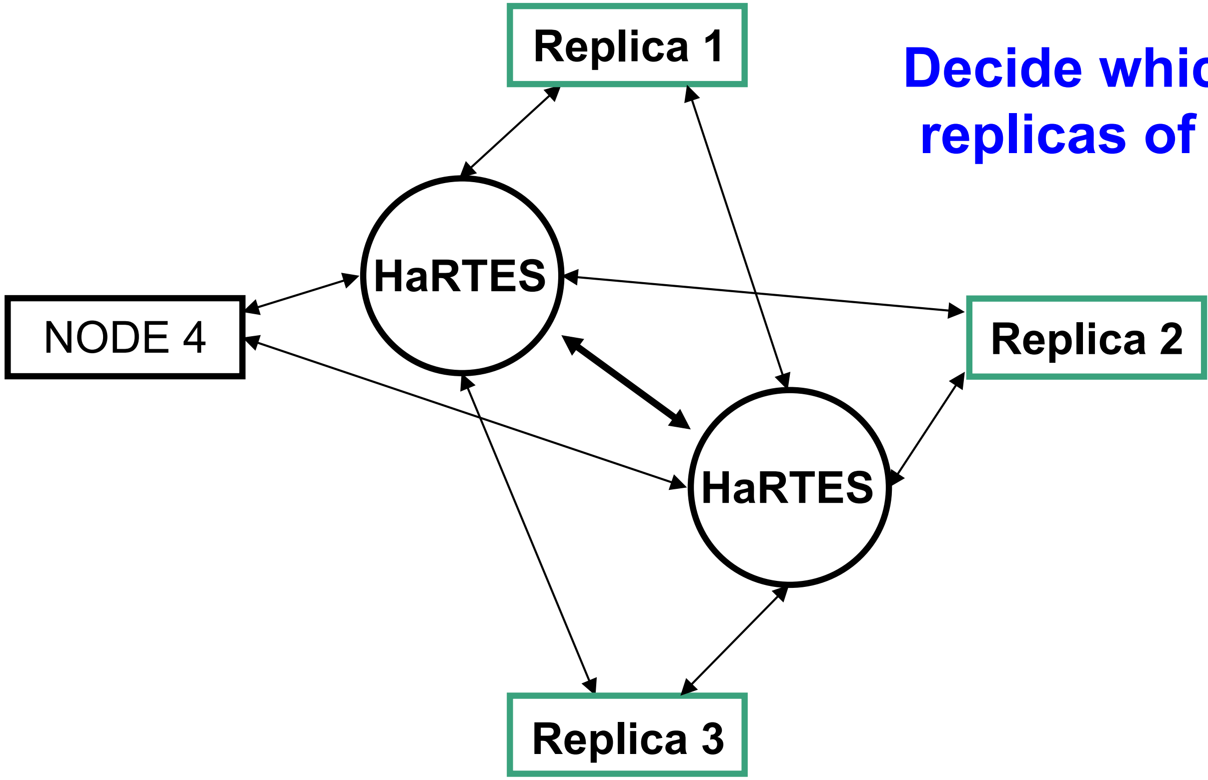
Dynamic Fault Tolerance

- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages



Dynamic Fault Tolerance

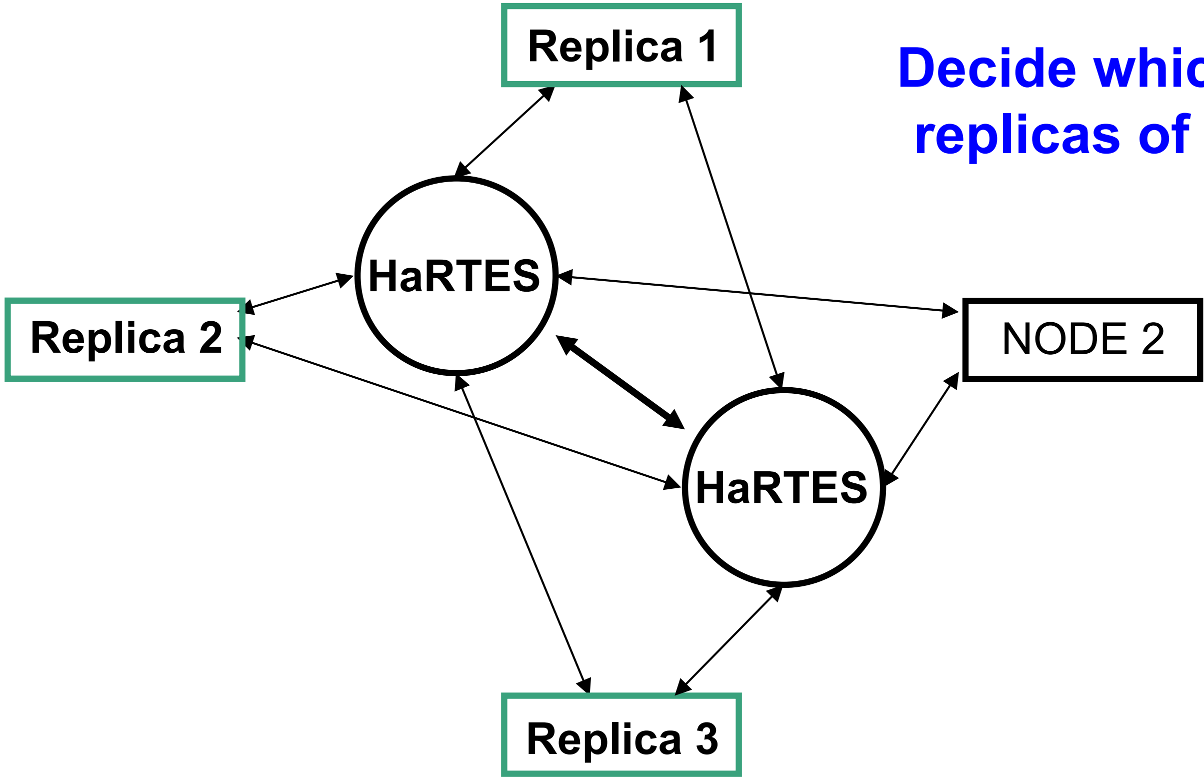
- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages



Decide which nodes act as replicas of a specific task

Dynamic Fault Tolerance

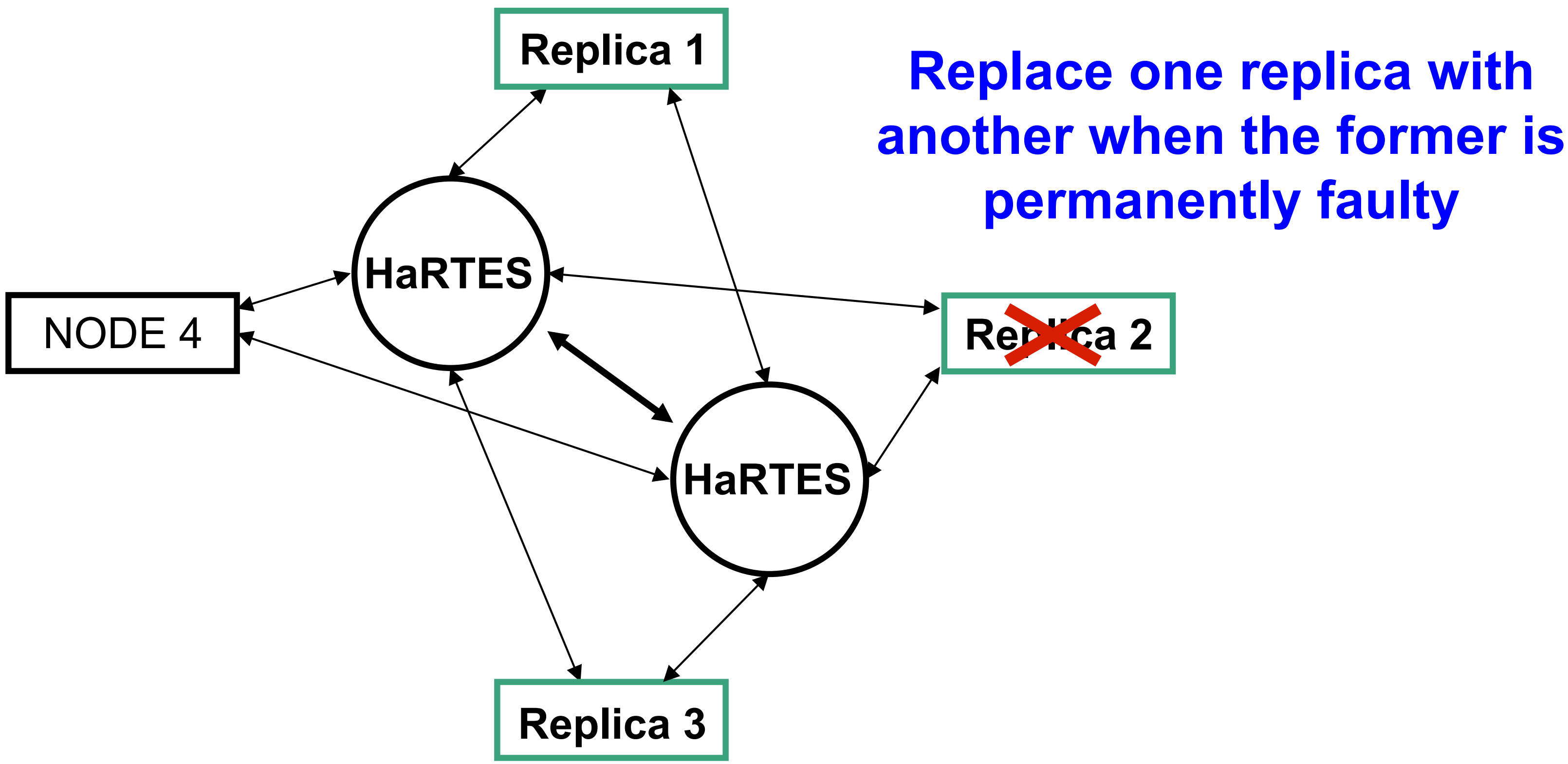
- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages



Decide which nodes act as replicas of a specific task

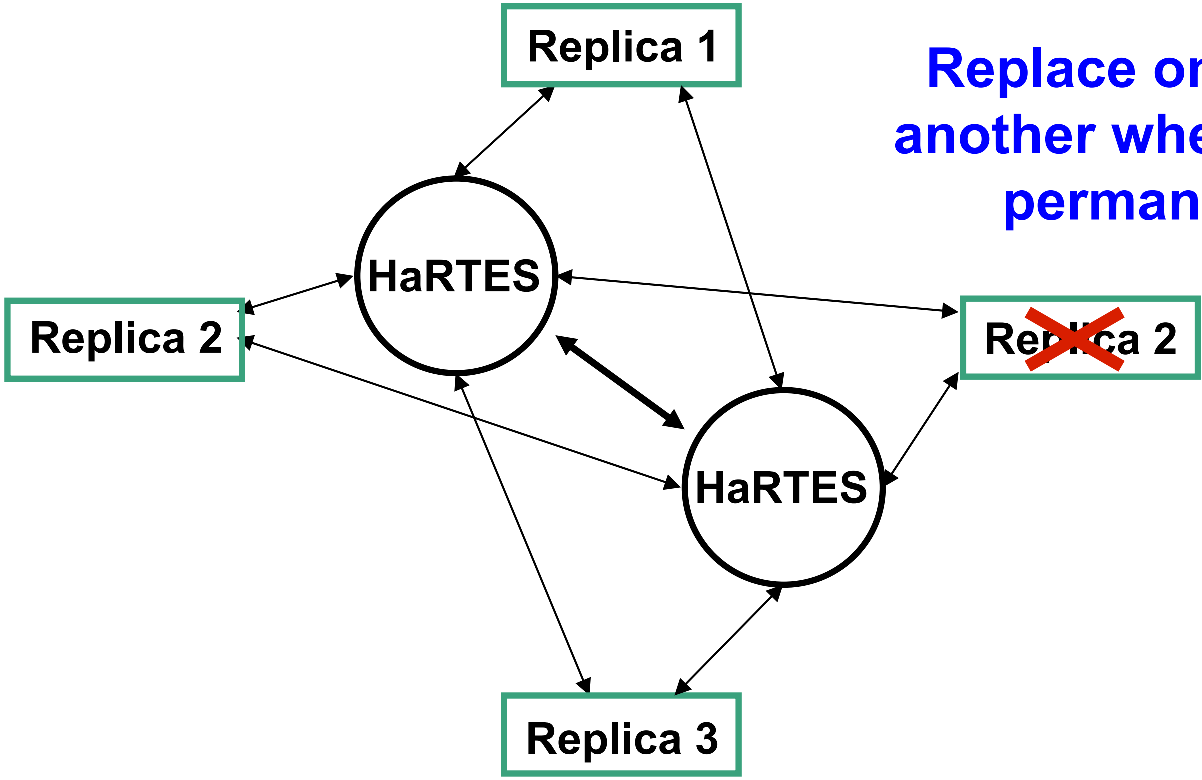
Dynamic Fault Tolerance

- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages



Dynamic Fault Tolerance

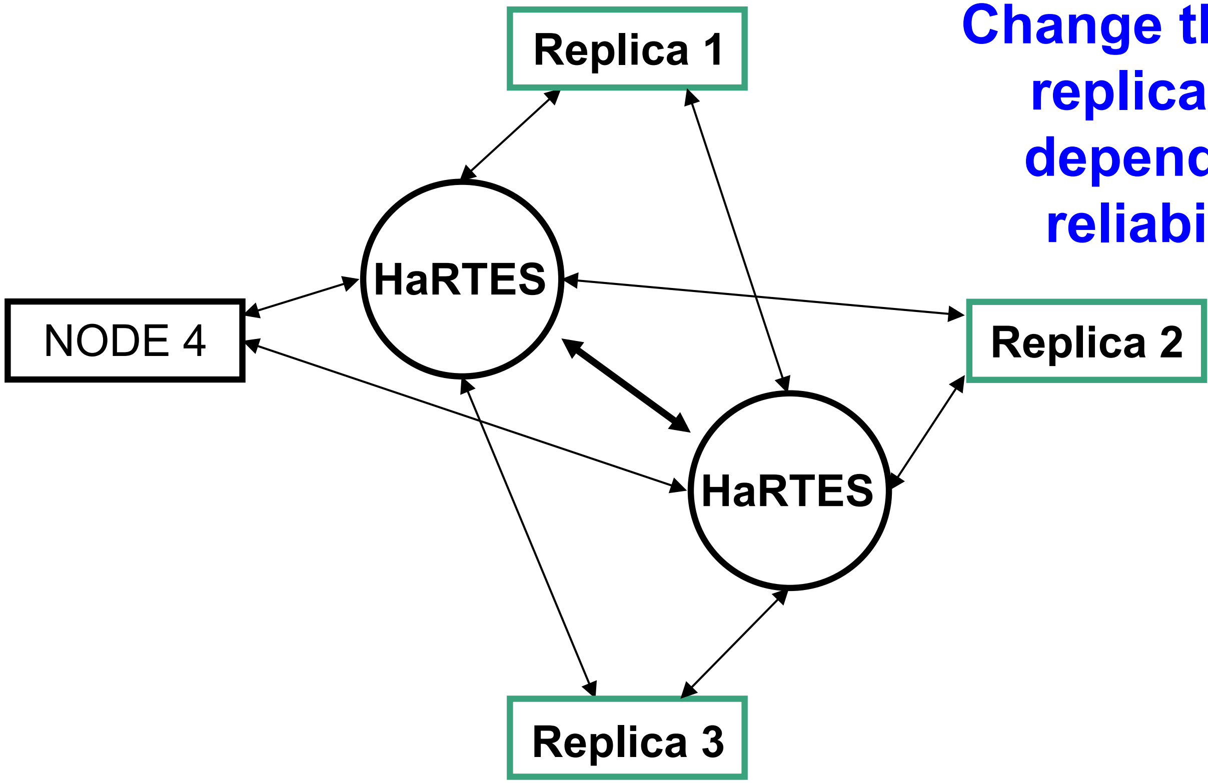
- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages



Replace one replica with another when the former is permanently faulty

Dynamic Fault Tolerance

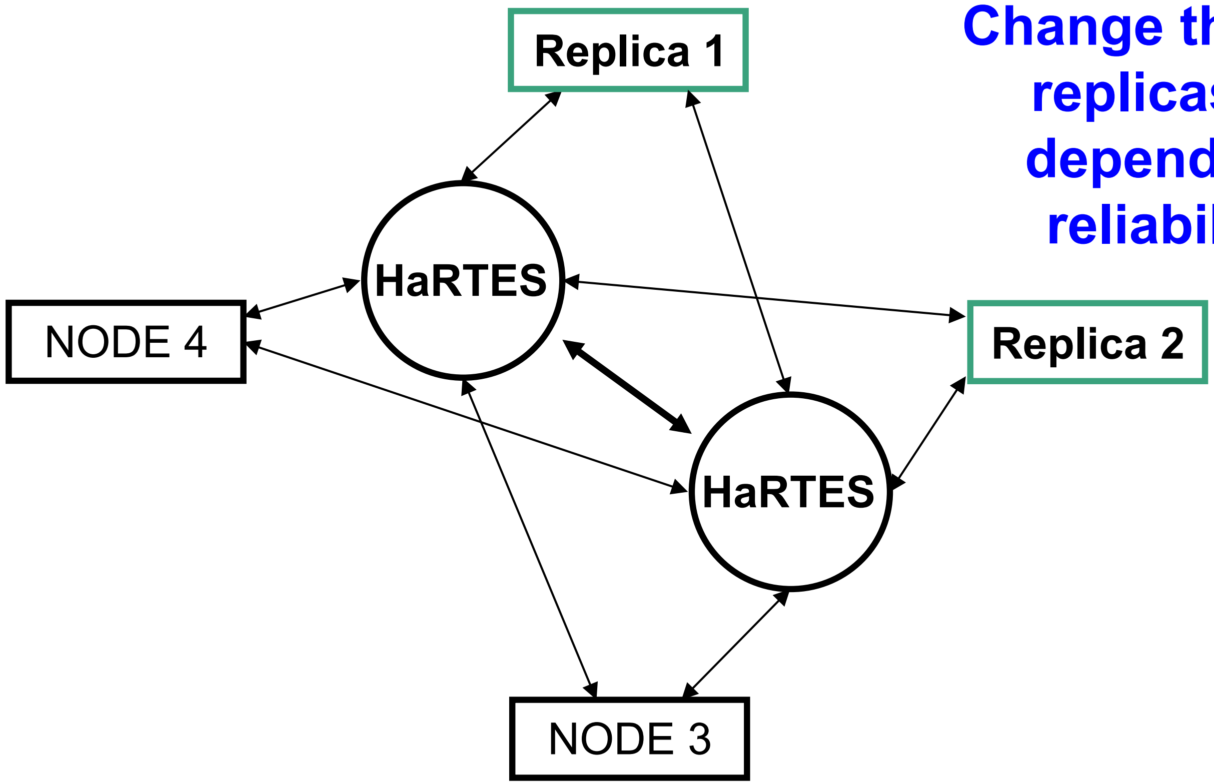
- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages



Change the number of replicas of a task depending on the reliability needs

Dynamic Fault Tolerance

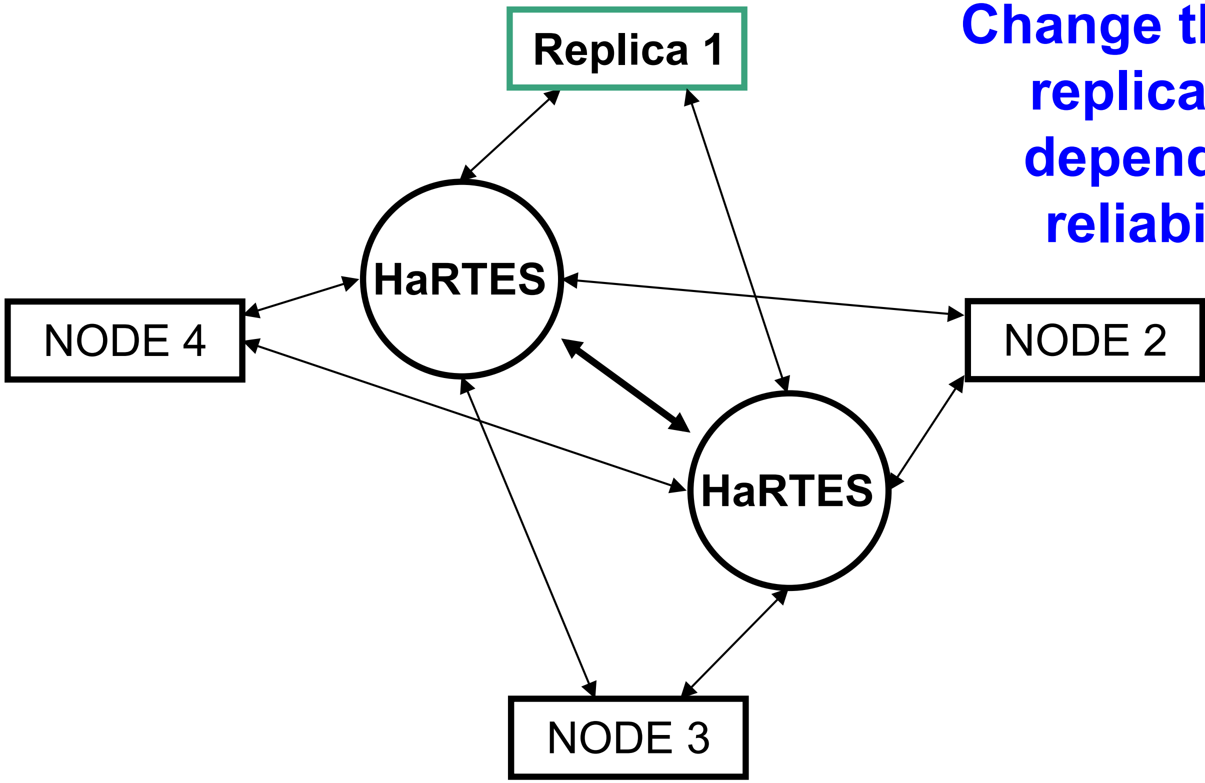
- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages



Change the number of replicas of a task depending on the reliability needs

Dynamic Fault Tolerance

- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages

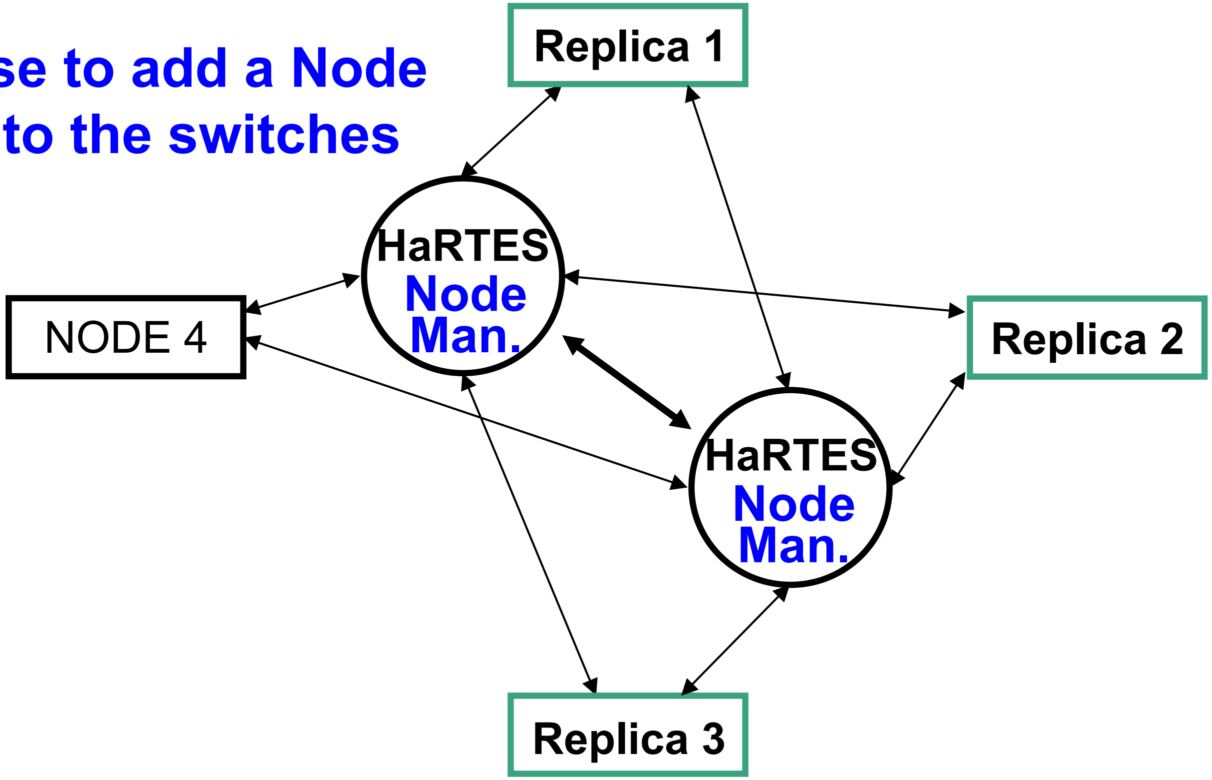


Change the number of replicas of a task depending on the reliability needs

Dynamic Fault Tolerance

- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages

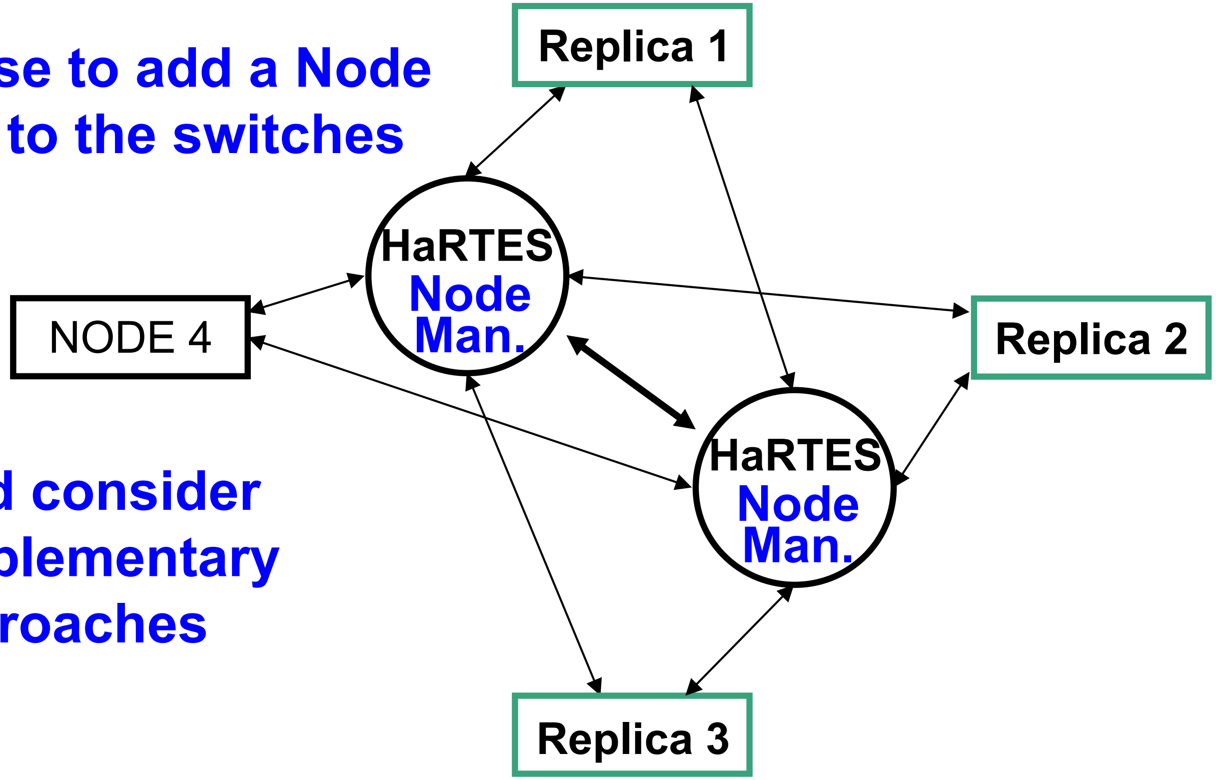
We propose to add a Node Manager to the switches



Dynamic Fault Tolerance

- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages

We propose to add a Node Manager to the switches

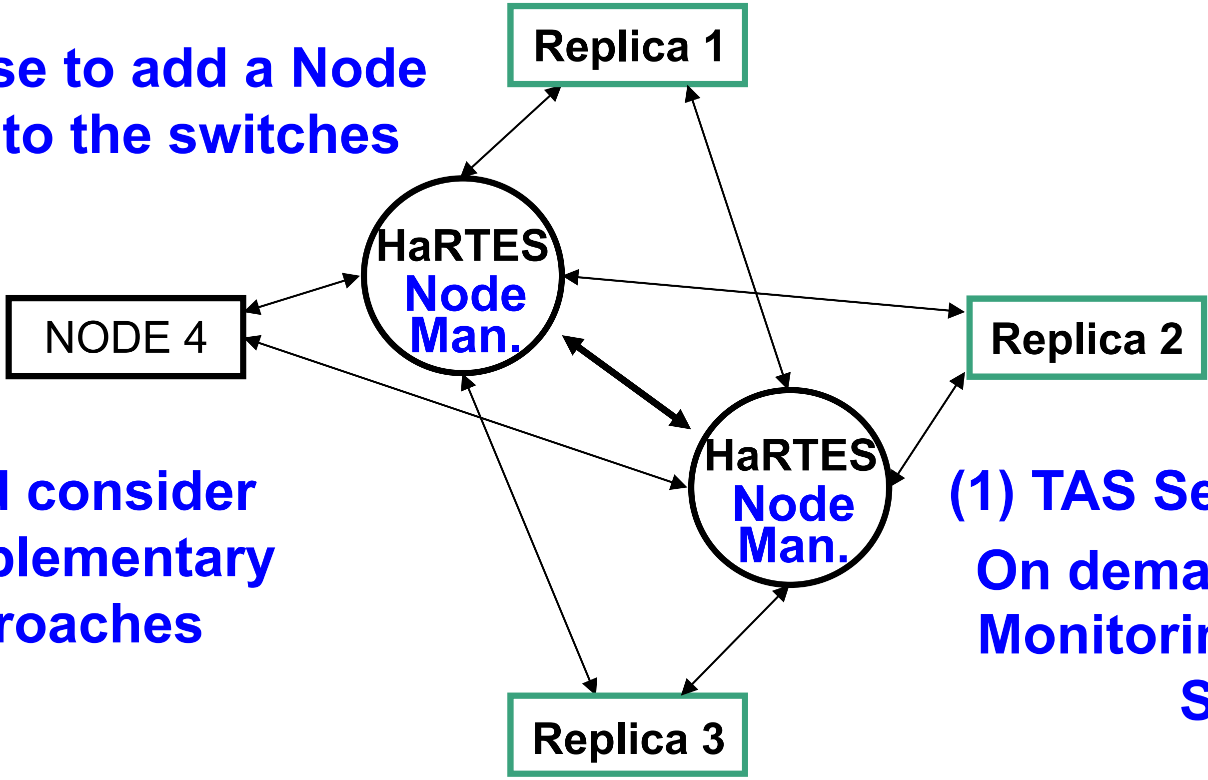


… and consider 3 complementary approaches

Dynamic Fault Tolerance

- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages

We propose to add a Node Manager to the switches



… and consider 3 complementary approaches

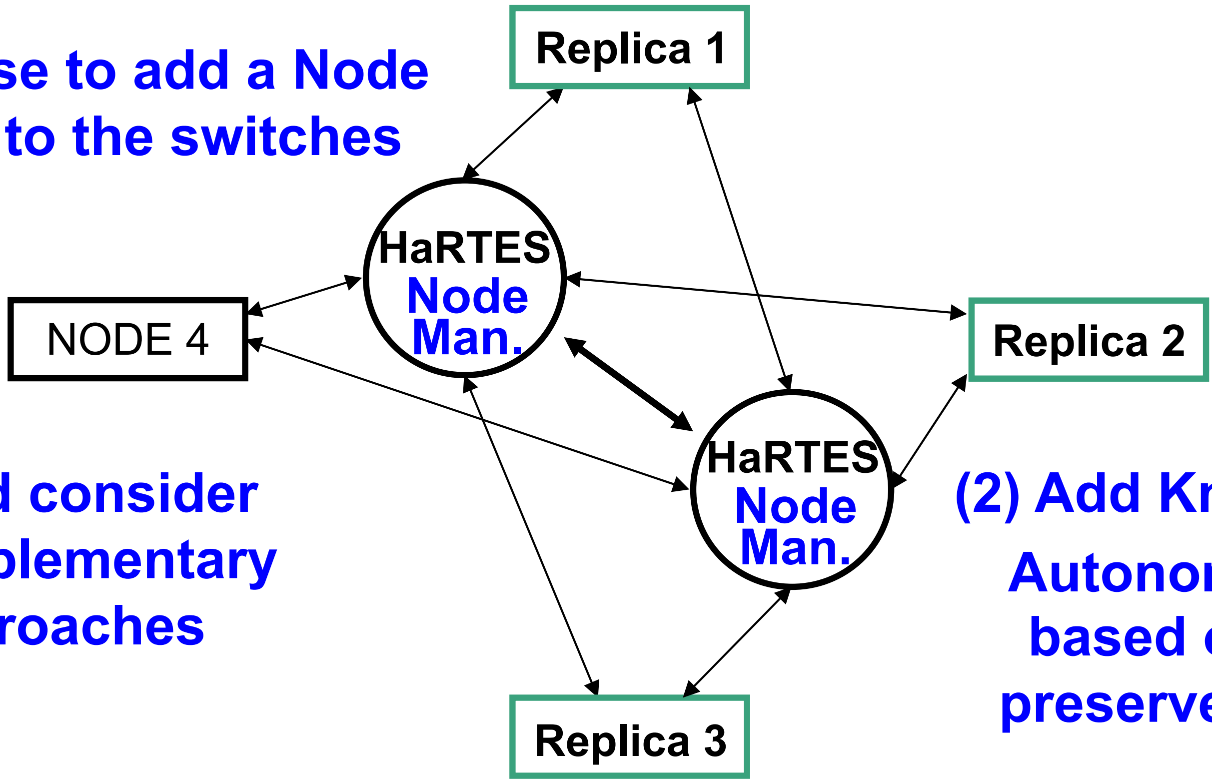
(1) TAS Services: FTT-like

On demand from nodes Monitoring and Reconf. Support

Dynamic Fault Tolerance

- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages

We propose to add a Node Manager to the switches



… and consider 3 complementary approaches

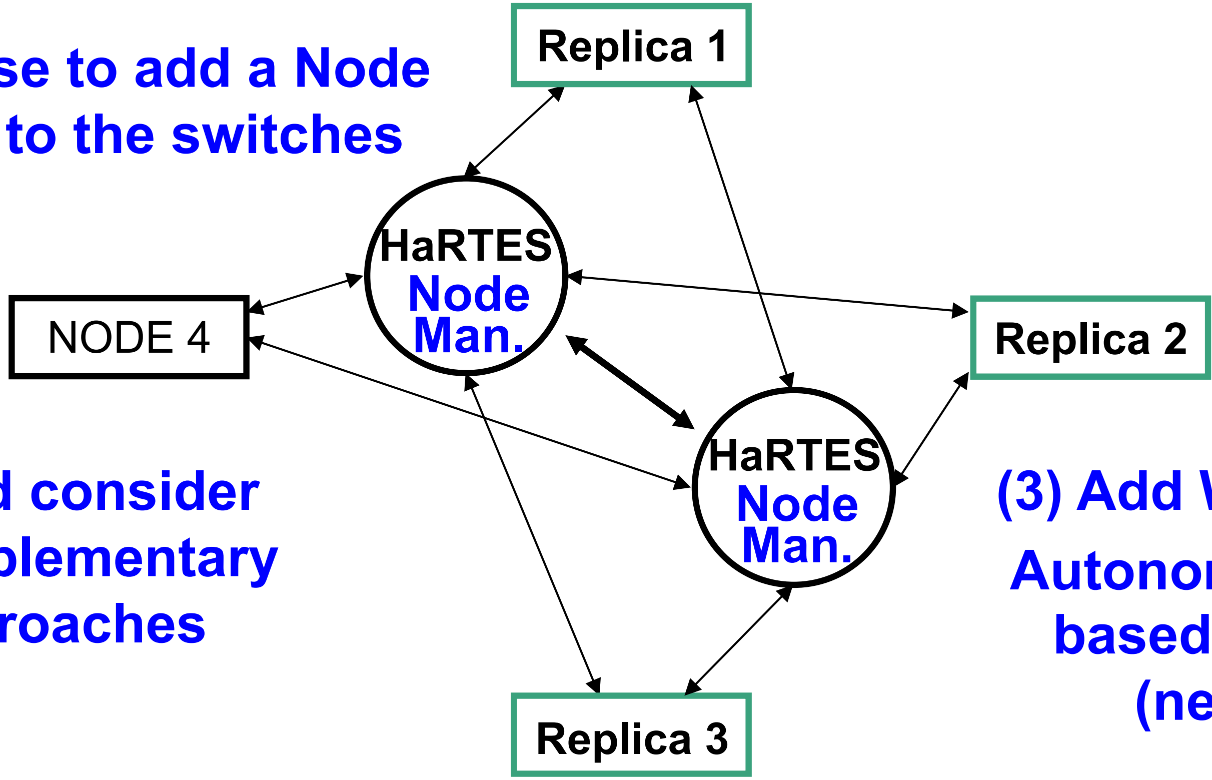
(2) Add Knowledge Entity

Autonomous Reconf. based on rules (e.g. preserve redundancy)

Dynamic Fault Tolerance

- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages

We propose to add a Node Manager to the switches



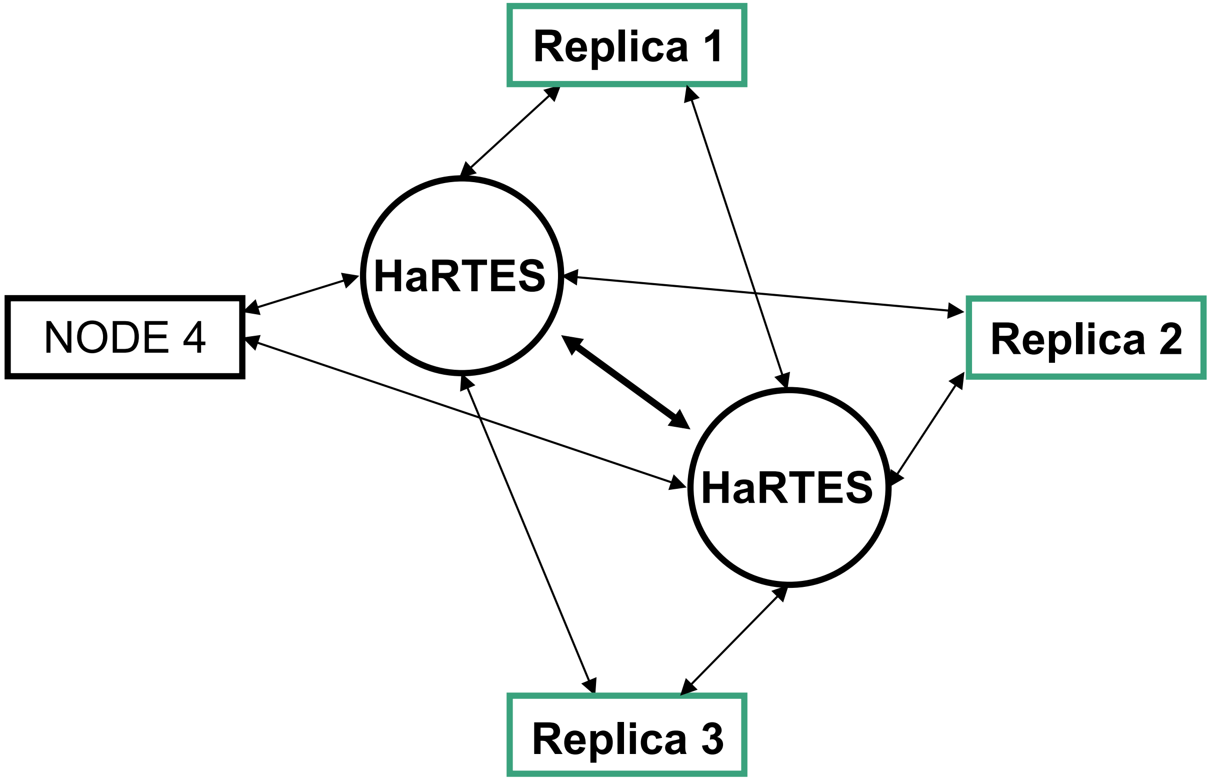
… and consider 3 complementary approaches

(3) Add Wisdom Entity

Autonomous Reconf. based on learning (new to us!)

Dynamic Fault Tolerance

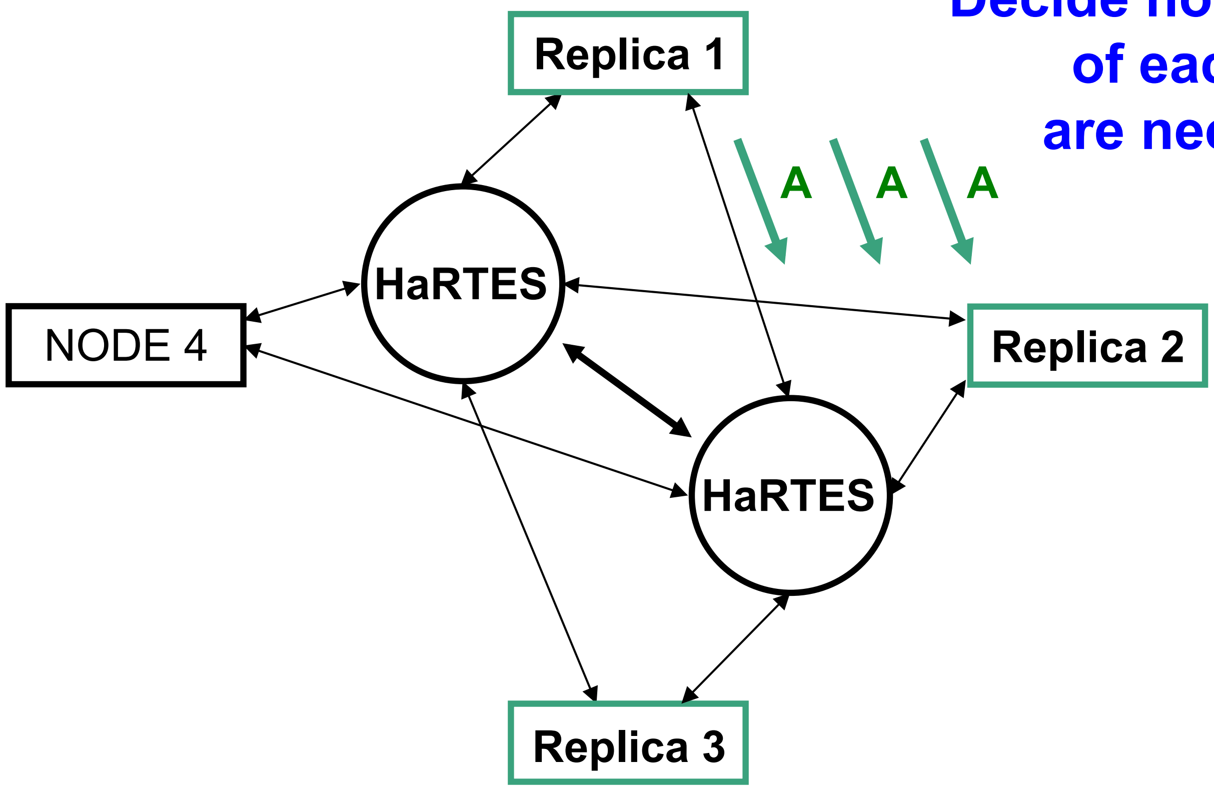
- **Targeted mechanisms**
  – Spatial replication of the nodes
  – Temporal replication of the messages



Dynamic Fault Tolerance

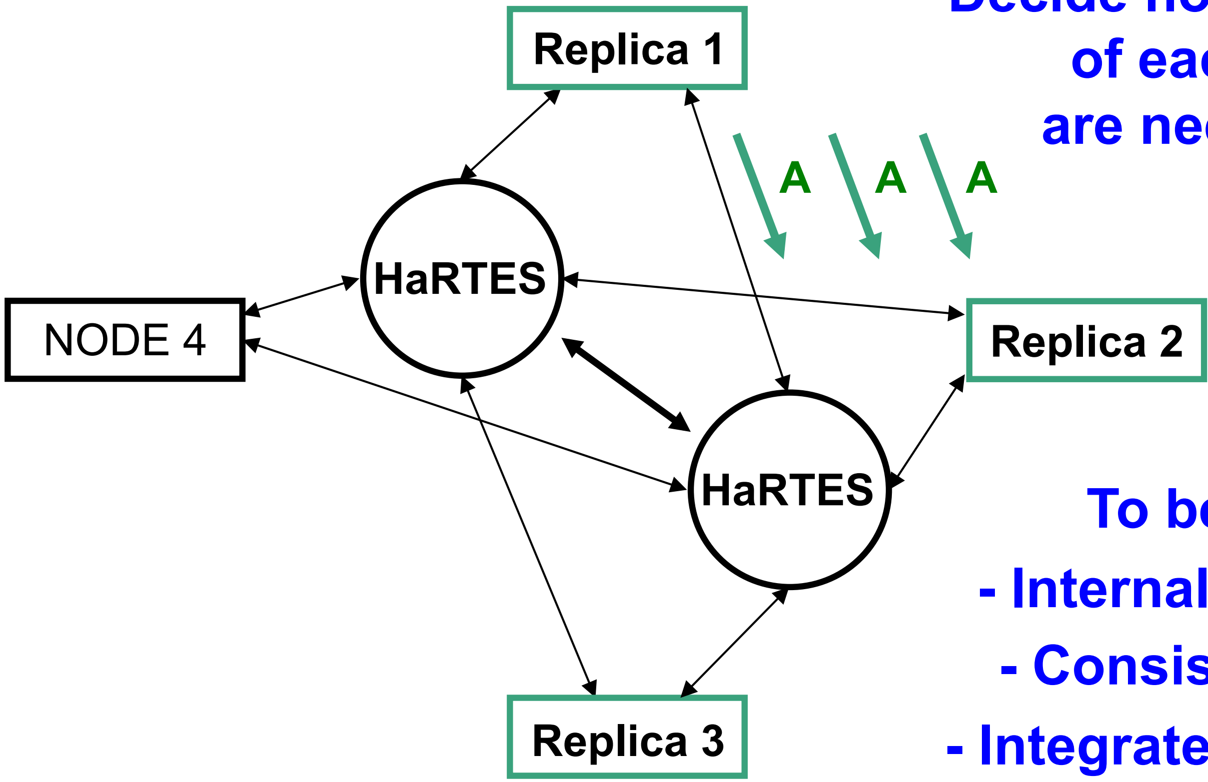
- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages

Decide how many replicas of each message are needed per link



Dynamic Fault Tolerance

- **Targeted mechanisms**
  - Spatial replication of the nodes
  - Temporal replication of the messages



Decide how many replicas of each message are needed per link

To be decided...

- Internal & external info

- Consistent decisions

- Integrate with NMs & FTT

Extending our results to other techs

Extending our results to other techs

- **Time Sensitive Networking (TSN) is set of standards (around 60 now!) with potential for adaptive systems**
- **Aims at providing standard Ethernet with additional services:**
  - Hard Real-Time
  - Reliability
  - Flexibility
  - Configurability
- **Internet of Things and Cyber-Physical Systems**
  - Automotive
  - Smart grid
  - Smart homes

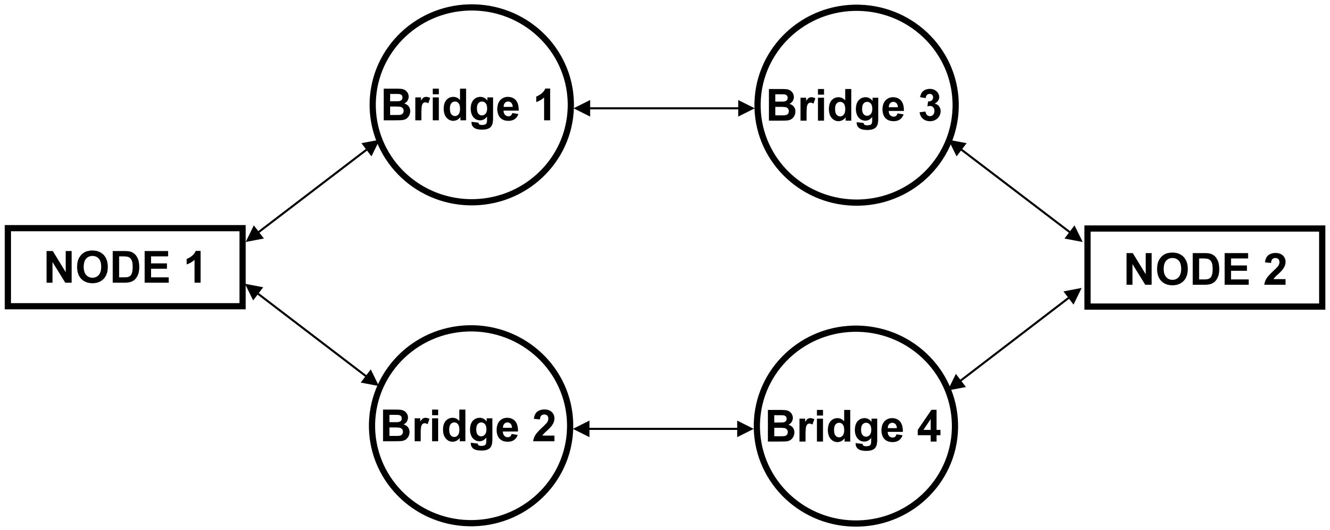
Extending our results to other techs

- **Our upcoming next project application is to be about TSN**

- **This is one of the aspects to be discussed in this meeting due to its potential for continuing our collaboration.**

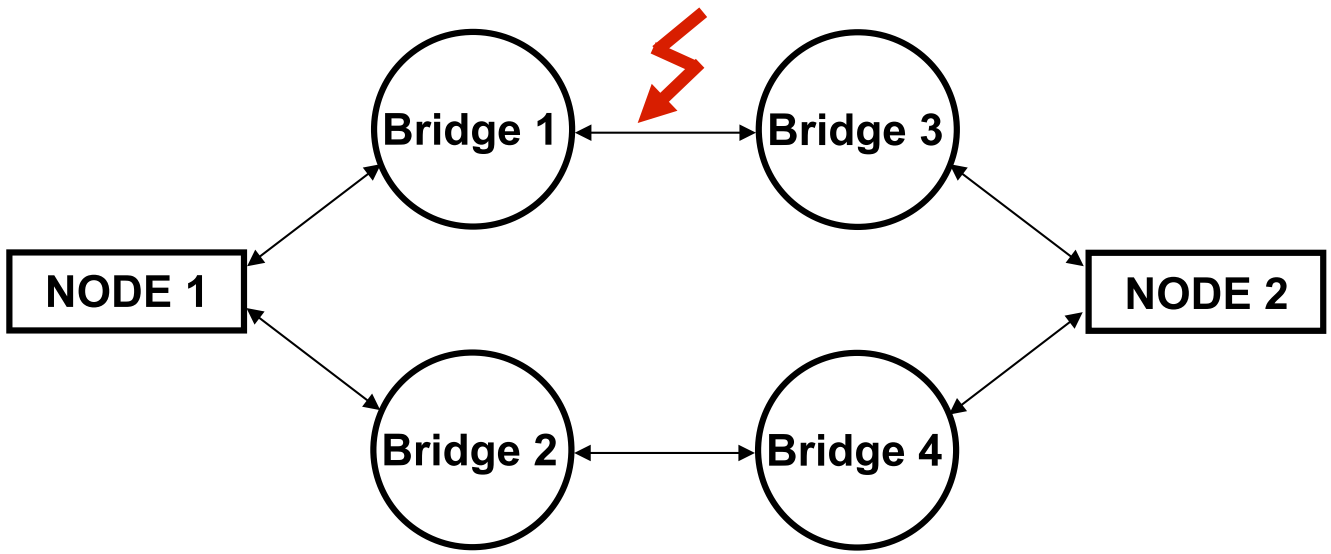
Extending our results to other techs

- **Time Sensitive Networking (TSN) is a set of standards with potential for adaptive systems**
  – We are now proposing a solution to the **tolerance to transient faults** in the comms without having to rely on spatial redundancy



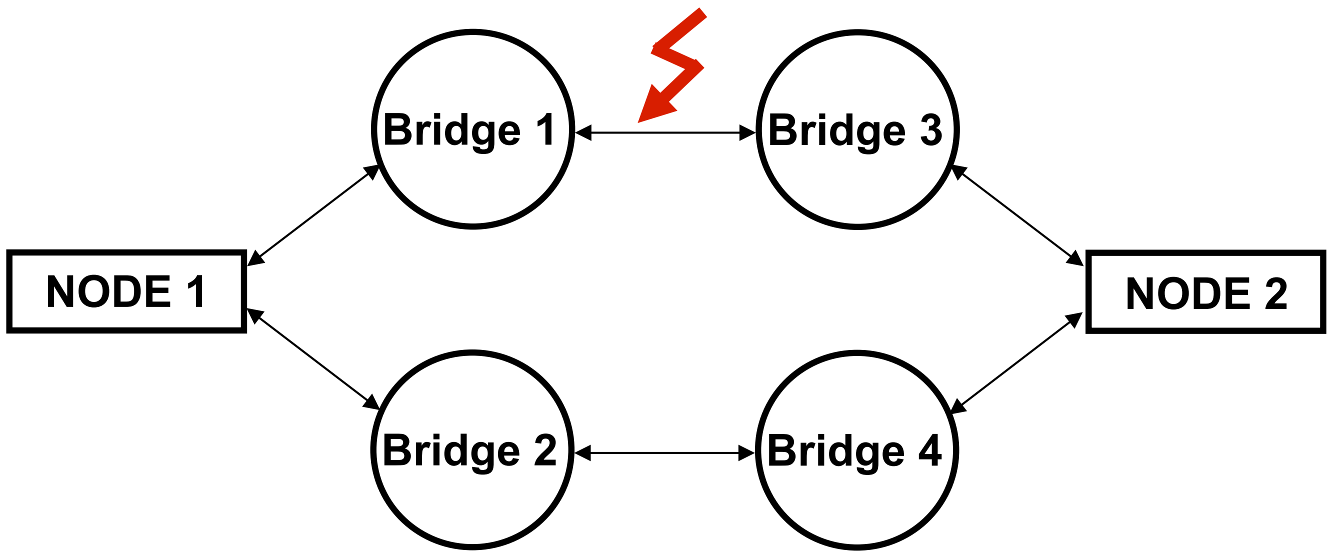
Extending our results to other techs

- **Time Sensitive Networking (TSN) is a set of standards with potential for adaptive systems**
  - We are now proposing a solution to the **tolerance to transient faults** in the comms without having to rely on spatial redundancy



Extending our results to other techs

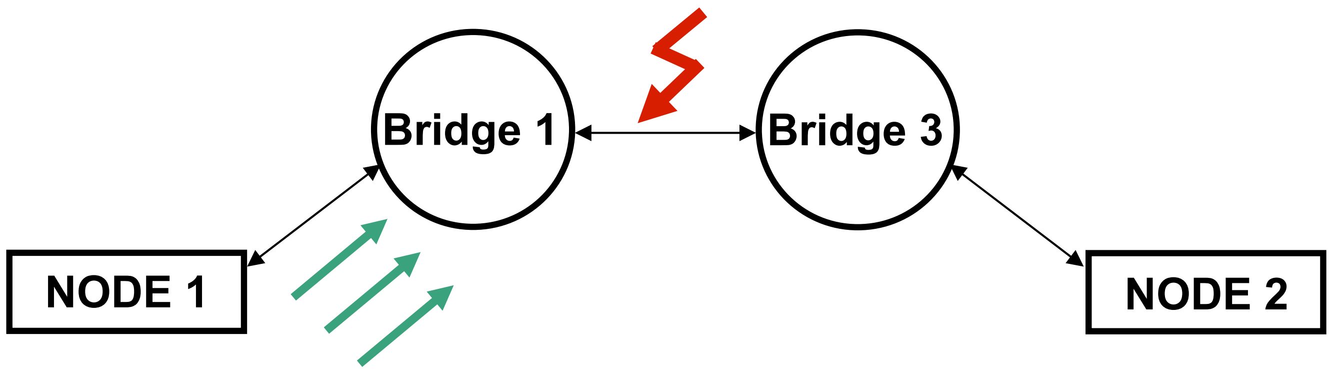
- **Time Sensitive Networking (TSN) is a set of standards with potential for adaptive systems**
  - We are now proposing a solution to the **tolerance to transient faults** in the comms without having to rely on spatial redundancy



What if spatial redundancy is not available any more?

Extending our results to other techs

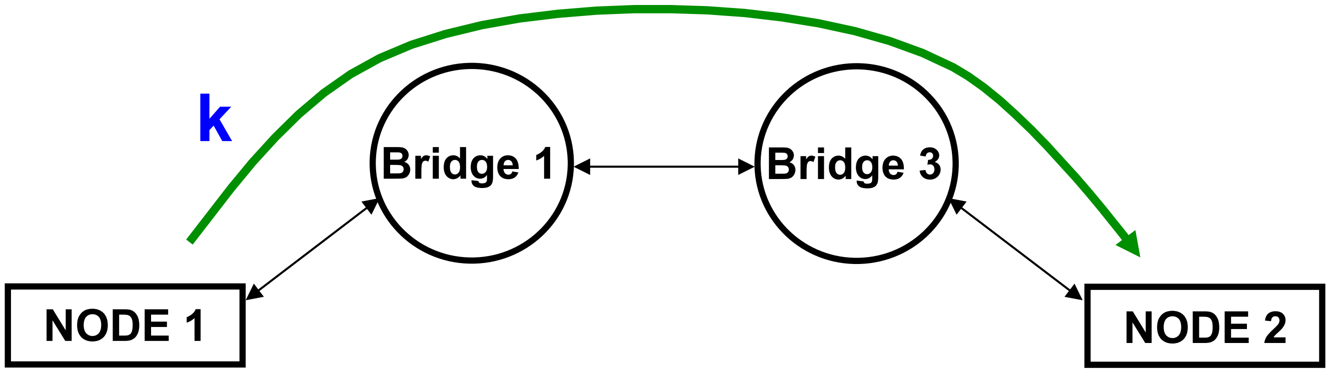
- **Time Sensitive Networking (TSN) is a set of standards with potential for adaptive systems**
  - We are now proposing a solution to the **tolerance to transient faults** in the comms without having to rely on spatial redundancy



Better to use time redundancy

Extending our results to other techs

- **Time Sensitive Networking (TSN) is a set of standards with potential for adaptive systems**
  - We are now proposing a solution to the **tolerance to transient faults** in the comms without having to rely on spatial redundancy

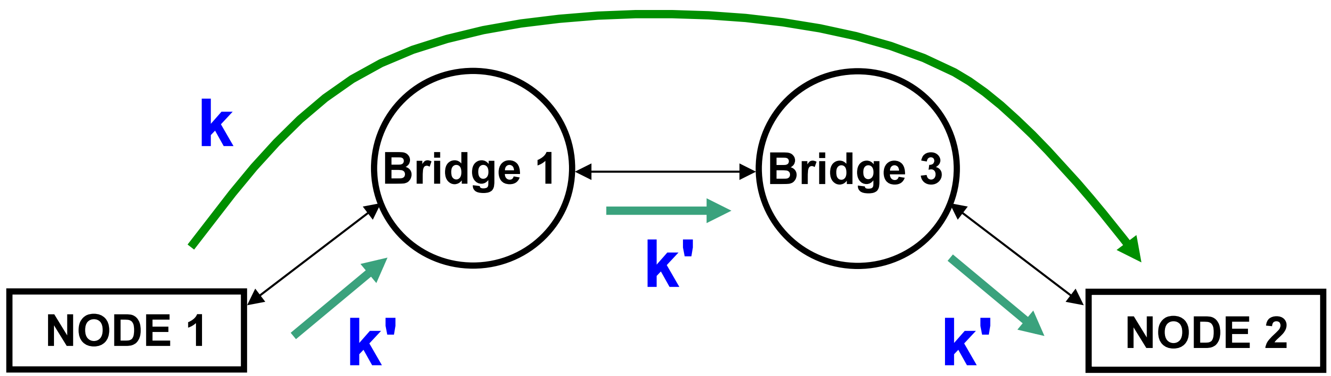


We have compared 2 approaches:

(1) Node 1 sends k replicas and each bridge forwards them all

Extending our results to other techs

- **Time Sensitive Networking (TSN) is a set of standards with potential for adaptive systems**
  - We are now proposing a solution to the **tolerance to transient faults** in the comms without having to rely on spatial redundancy



We have compared 2 approaches:

(1) Node 1 sends k replicas and each bridge forwards them all
(2) Node 1 sends k' replicas and each bridge sends k' replicas of the first one it receives

Extending our results to other techs

- **Time Sensitive Networking (TSN) is a set of standards with potential for adaptive systems**
  - We are now proposing a solution to the **tolerance to transient faults** in the comms without having to rely on spatial redundancy

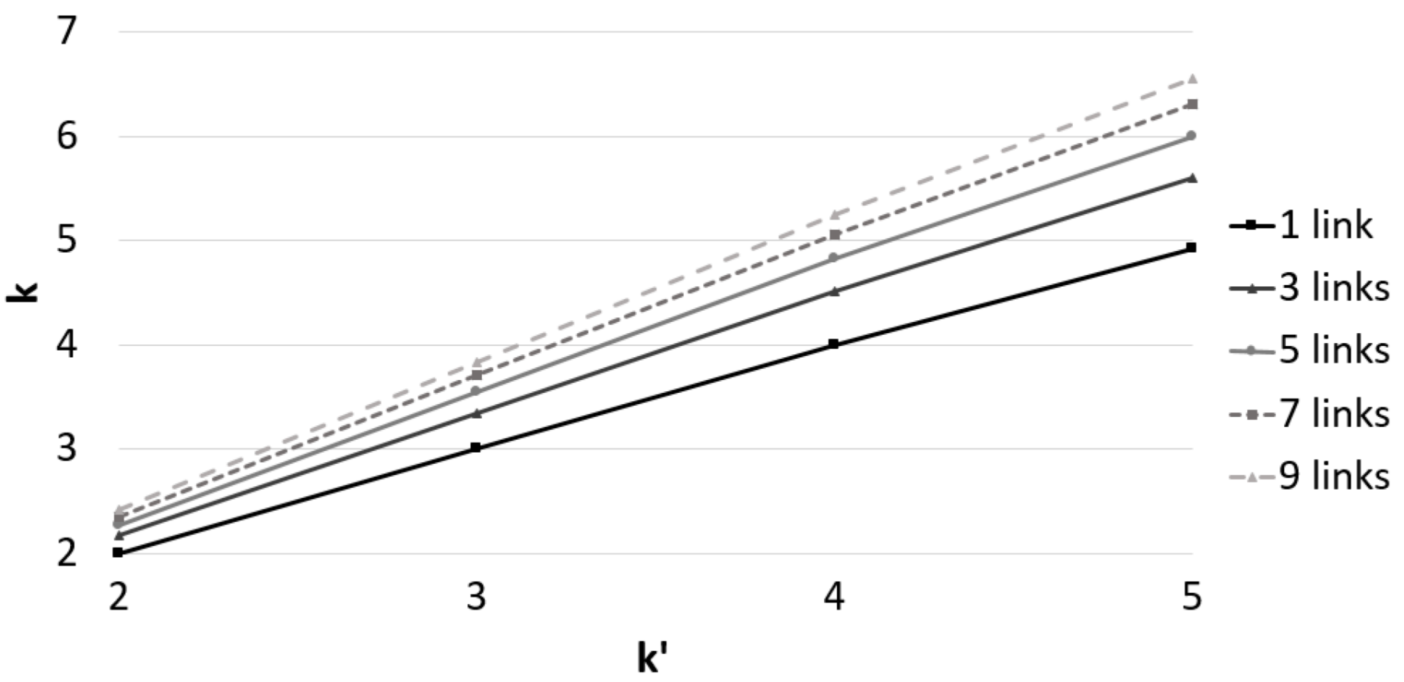


Fig. 2: k vs. k' for different number of links.

Summary

- In the previous project FT4FTT we introduced **static FT mechanism** to increase the reliability of FTT-based (Ethernet) systems

- Taking FT4FTT as starting point **we are introducing dynamic FT** in order to obtain a more adaptable system

- We are investigating in the addition of **dynamism to node replication and message replication**

- We are also considering to **extend our results to TSN** and to develop FT mechanisms specially tailored to the TSN characteristics (e.g. multi-hop topologies).

DFT4FTT: Dynamic FT for increasing the adaptivity of highly-reliable distributed embedded systems based on Flexible Time-Triggered Ethernet

Julián Proenza

Systems, Robotics and Vision Group. UIB.
SPAIN

UIB Universitat de les Illes Balears

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD