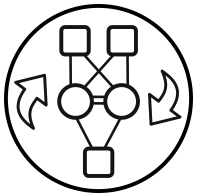
DFT4FTT: Dynamic FT for increasing the adaptivity of highly-reliable distributed embedded systems based on Flexible Time-Triggered Ethernet

Julián Proenza

Systems, Robotics and Vision Group. UIB.
SPAIN

Universitat de les Illes Balears



DTF4FTT Project Data

- **DTF4FTT**
  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
  - 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 team

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

The team

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

The team

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

The team

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



The team

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




The team

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





The team

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






The team

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







The team

- **UIB**
  - Manuel Barranco
  - Ignasi Furió
  - Pere Palmer
  - David Gessner
  - Sinisa Djerasevic
  - Alberto Ballesteros (PhD thesis)
  - **Inés Álvarez** (PhD thesis)
  - Julián Proenza

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

The team

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

The team

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

The team

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




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.

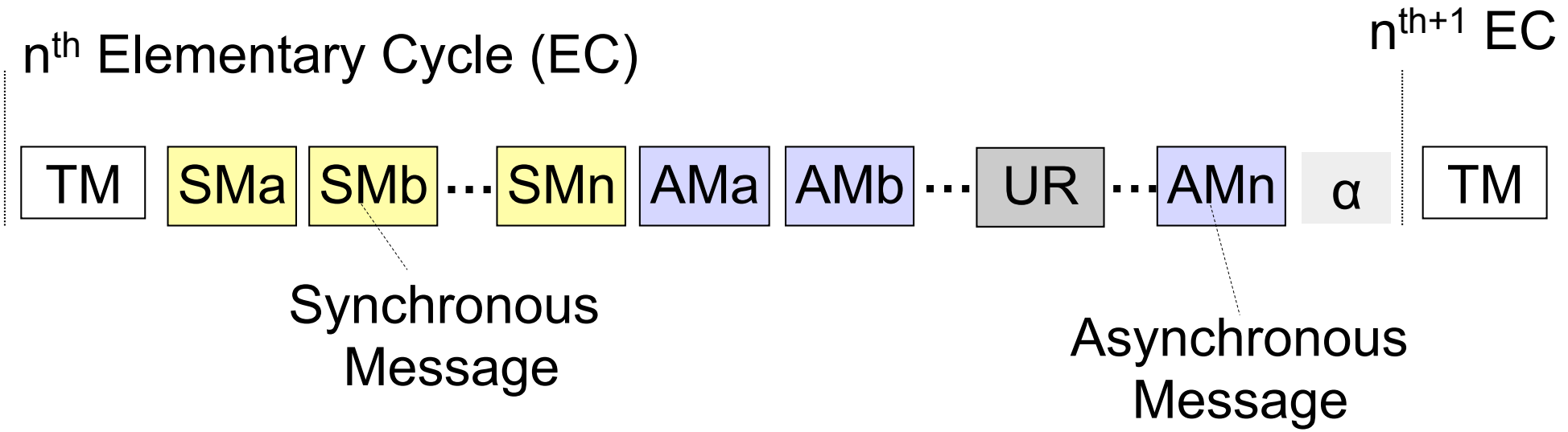
- Such systems are called adaptive embedded systems, and require services supporting…
  - flexibility, real-time and dependability at different levels of the system architecture, such as the OS and **the network**.

Context of the project Flexibility in FTT

- FTT is a very promising 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**

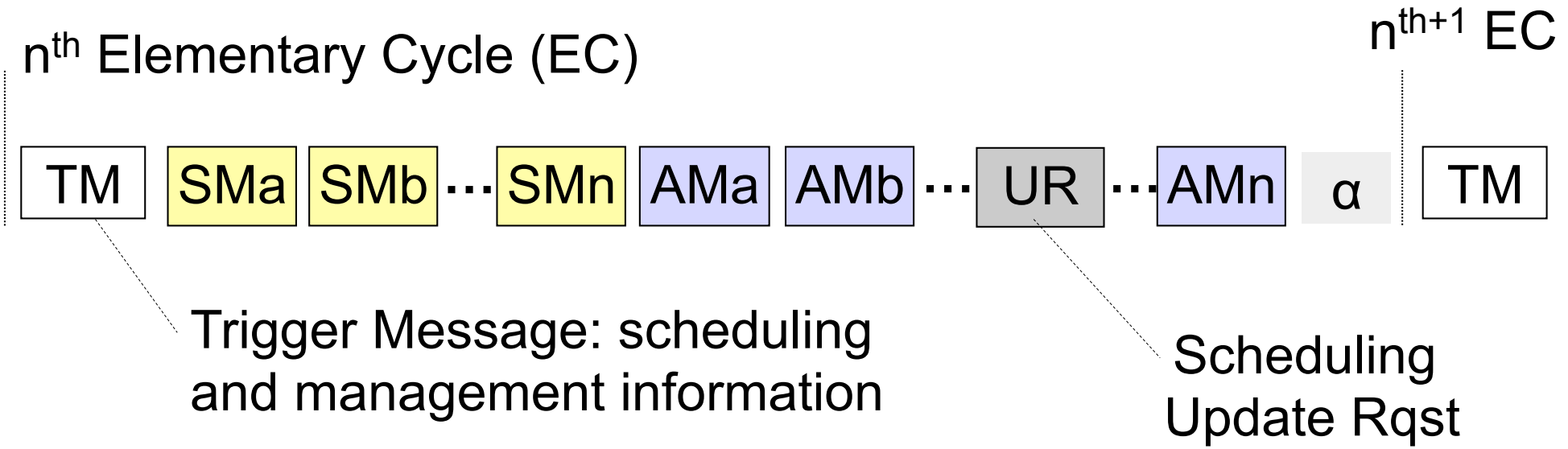
Context of the project
Flexibility in FTT

- 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**.

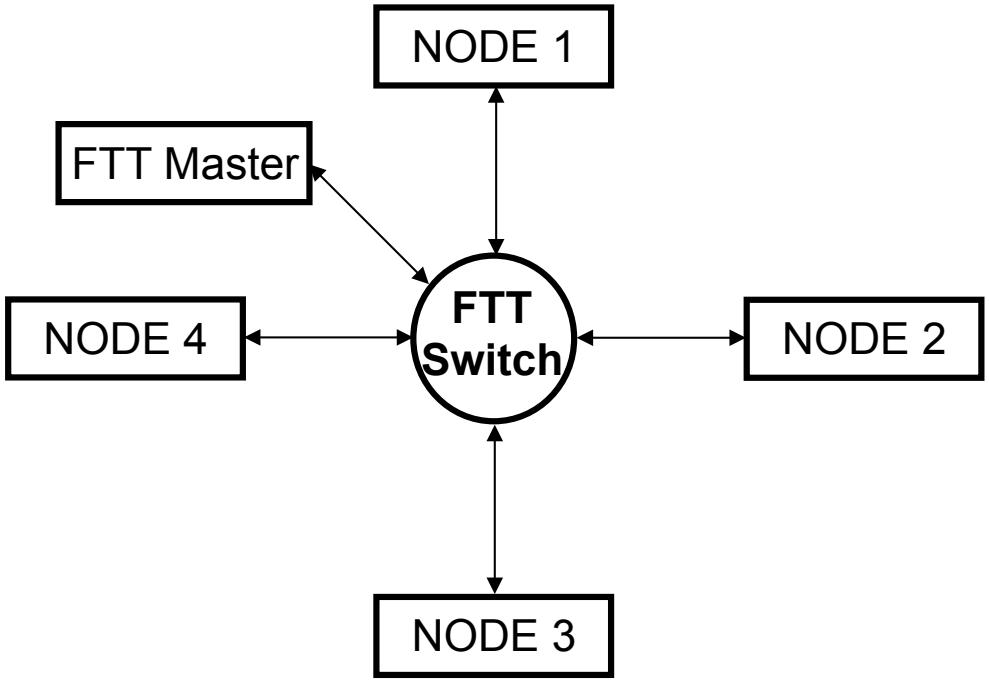


Context of the project FTT-Ethernet

- **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**

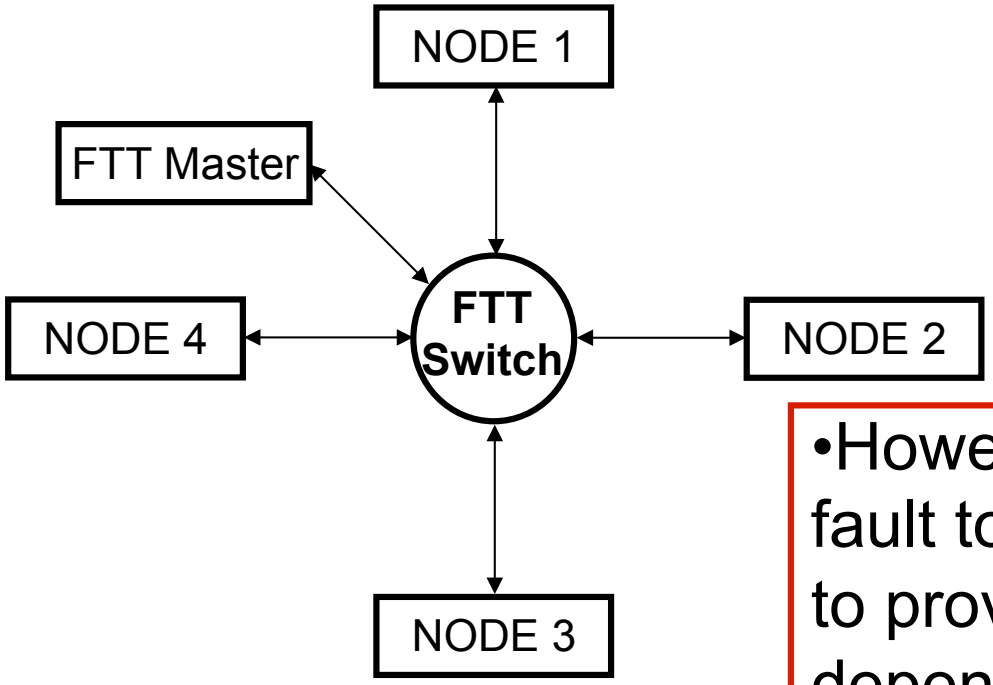
Context of the project FTT-Ethernet

- **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

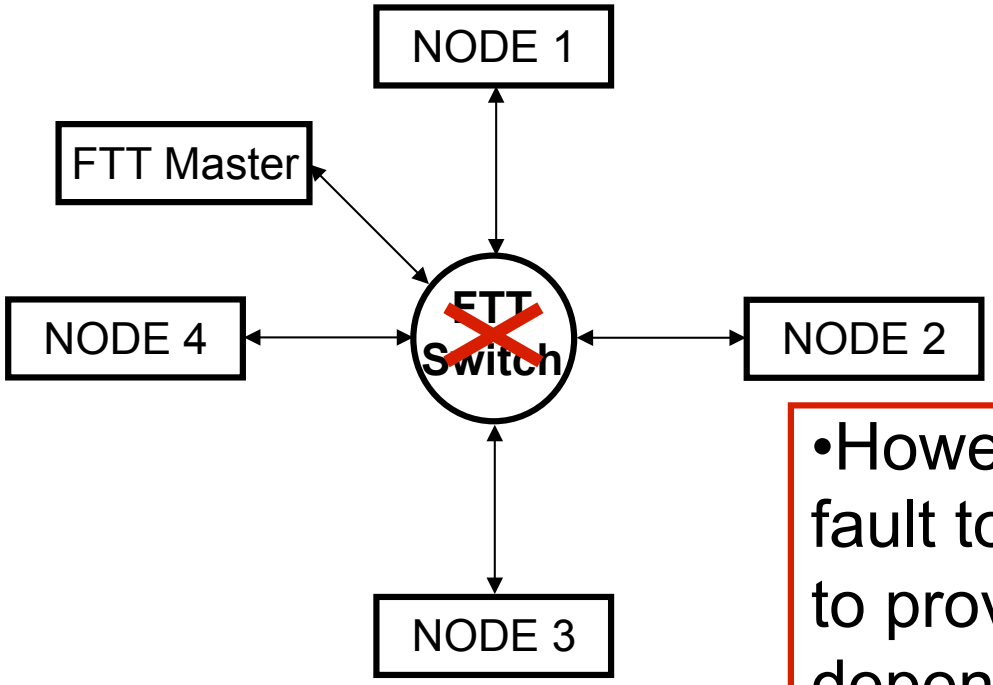
- **FTT-Ethernet** is the result of using FTT over the appealing Ethernet with RT response,
  - A higher potential thanks to the increase in bandwidth
  - **An FTT Switch (HaRTES) allows using legacy nodes**



•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 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.**

Specific objectives of FT4FTT

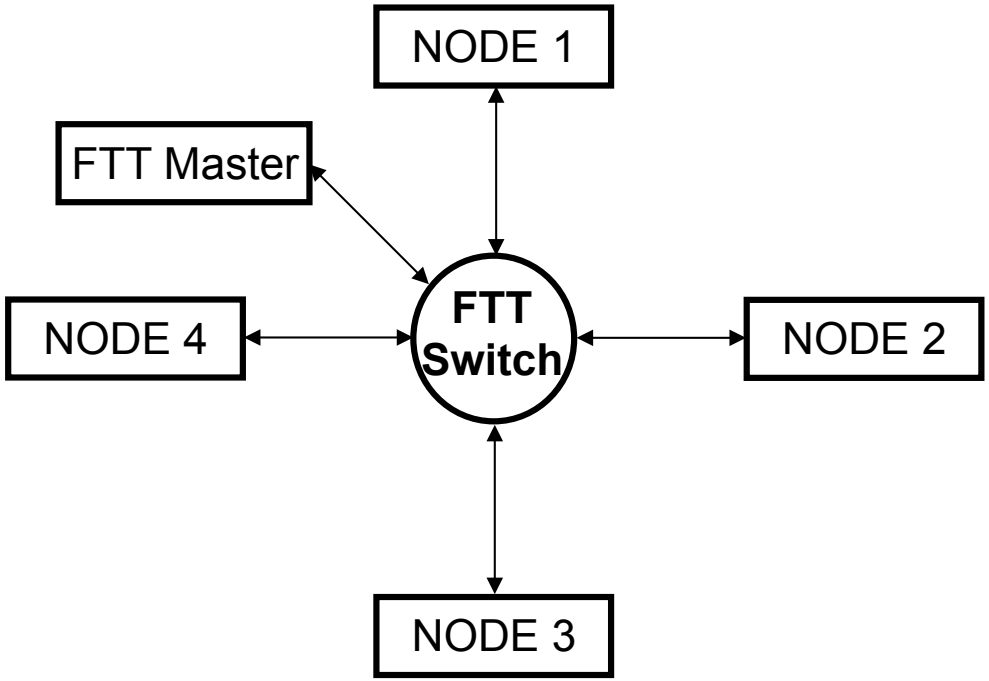
- 1) Achieve an increasing level of dependability for Ethernet infrastructures based on an FTTEnabled Switch, by means of the **incorporation of** basic **fault tolerance mechanisms**;

- 2) Thoroughly **evaluate the correctness** of the design as well as the achieved **level of dependability**;

- 3) **Develop a prototype** of said infrastructure in order to obtain experimental results and thus validate the whole infrastructure proposed.

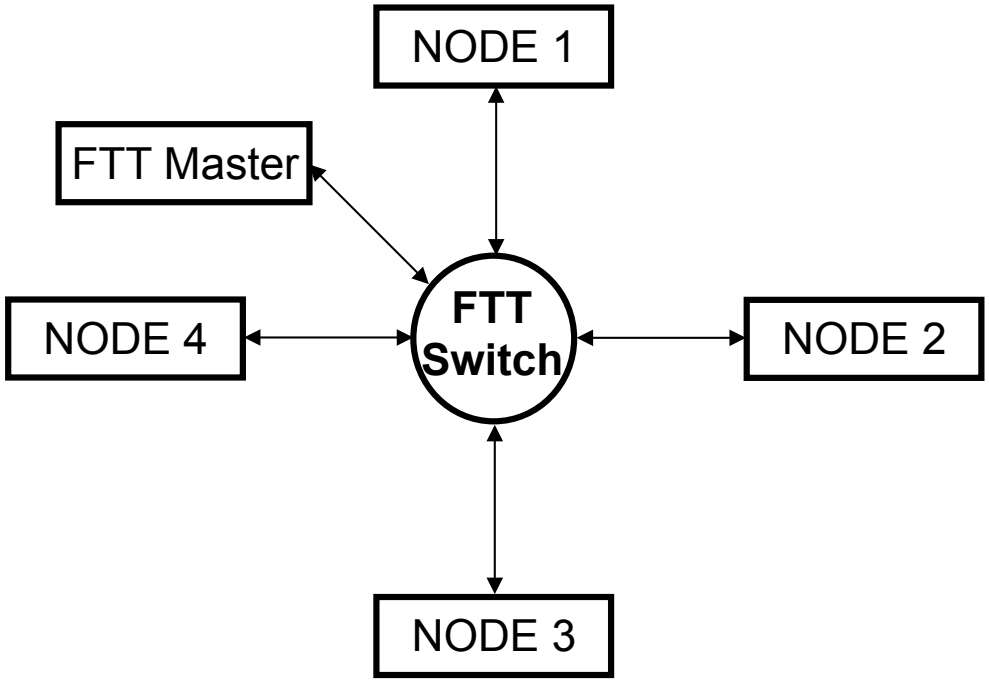
Our starting point

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



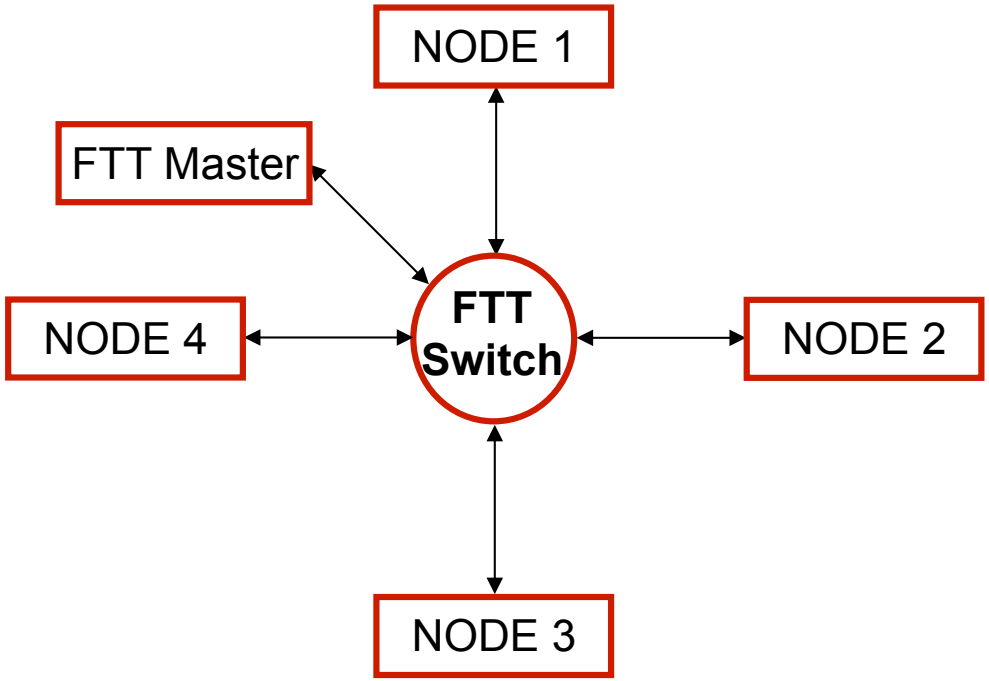
Our approach

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



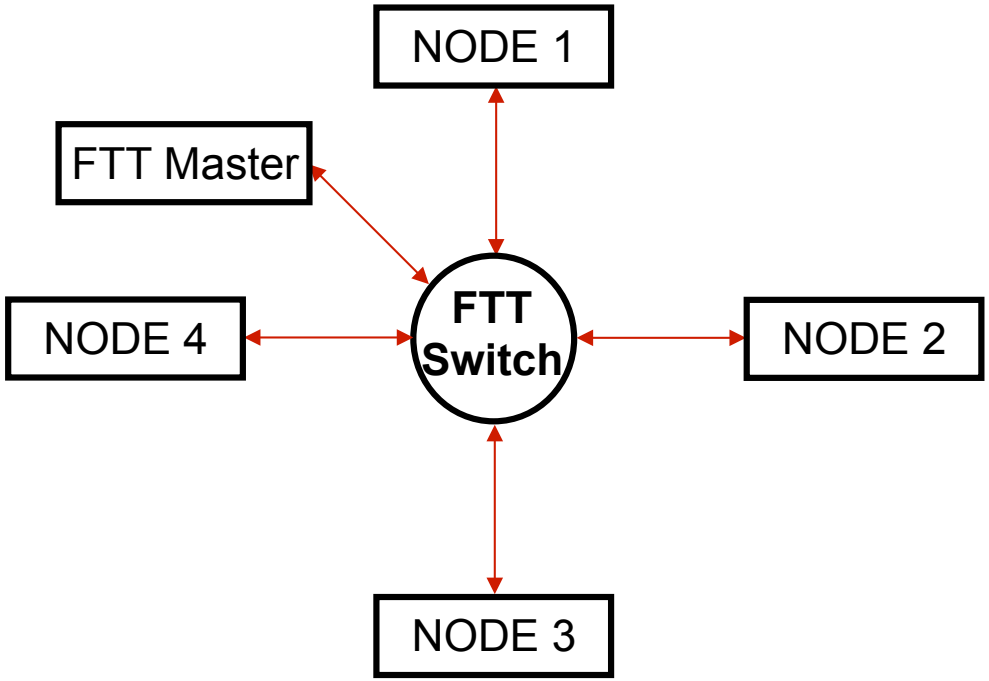
Fault model (1)

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



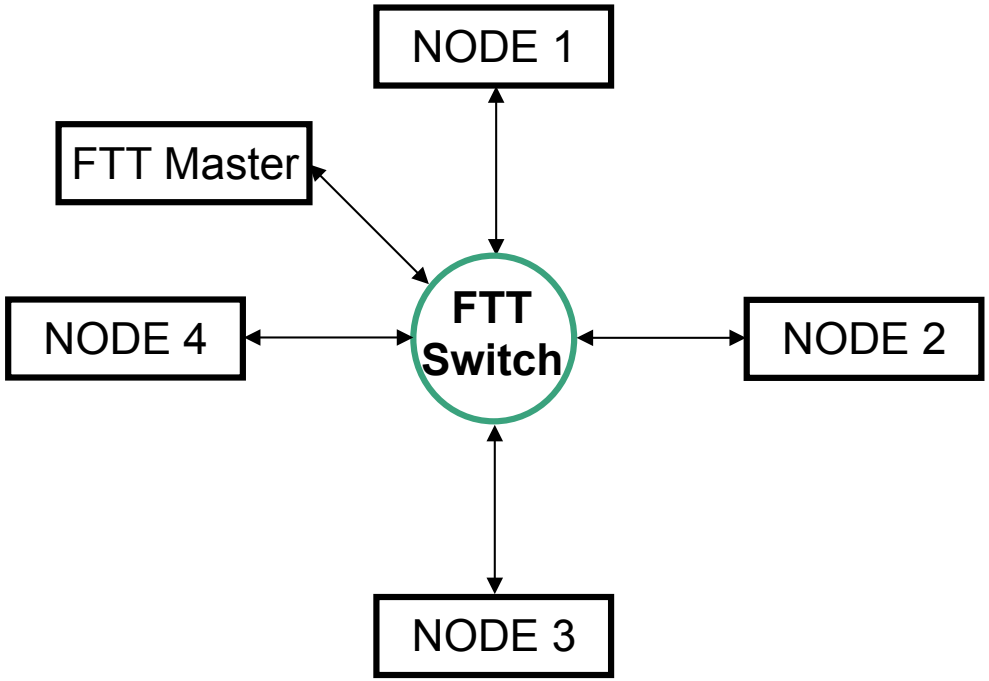
Fault model (2)

- Permanent and temporary faults in the links



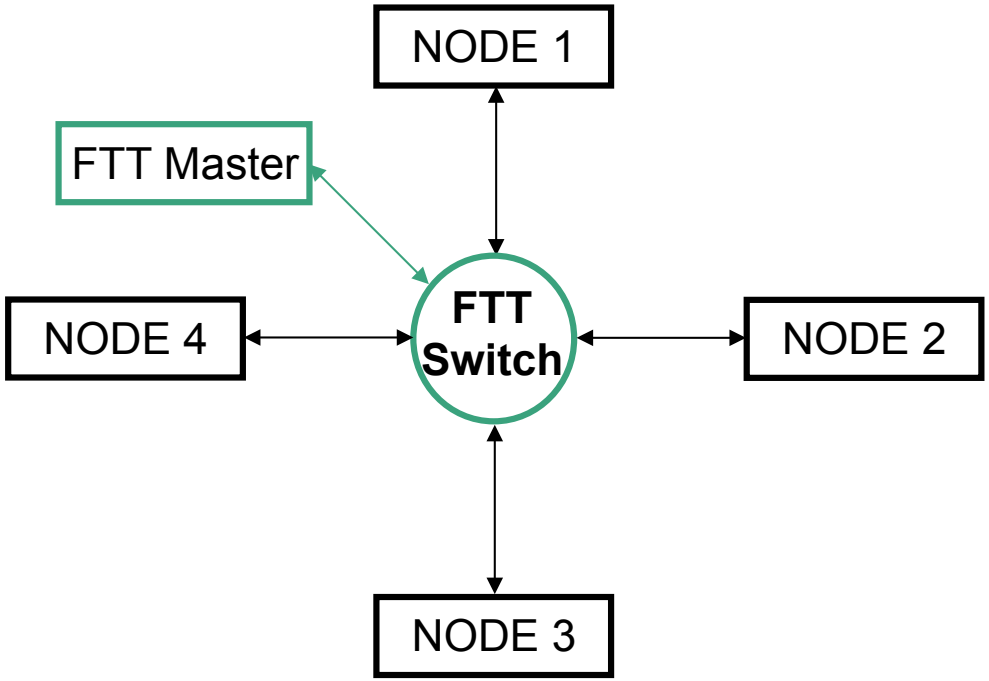
Our strategy (1)

- 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 an even legacy nodes**
  - to have a direct link between master and port guardians



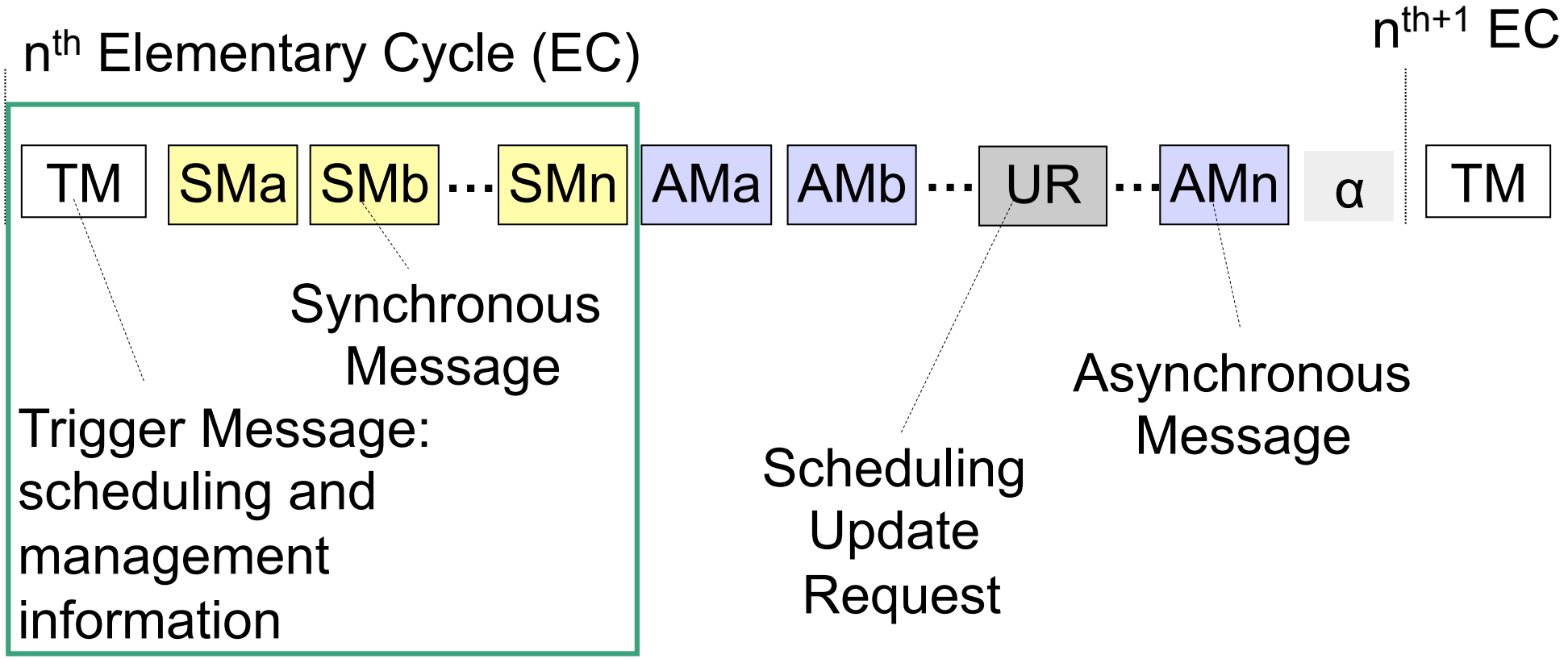
Our strategy (2)

- 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



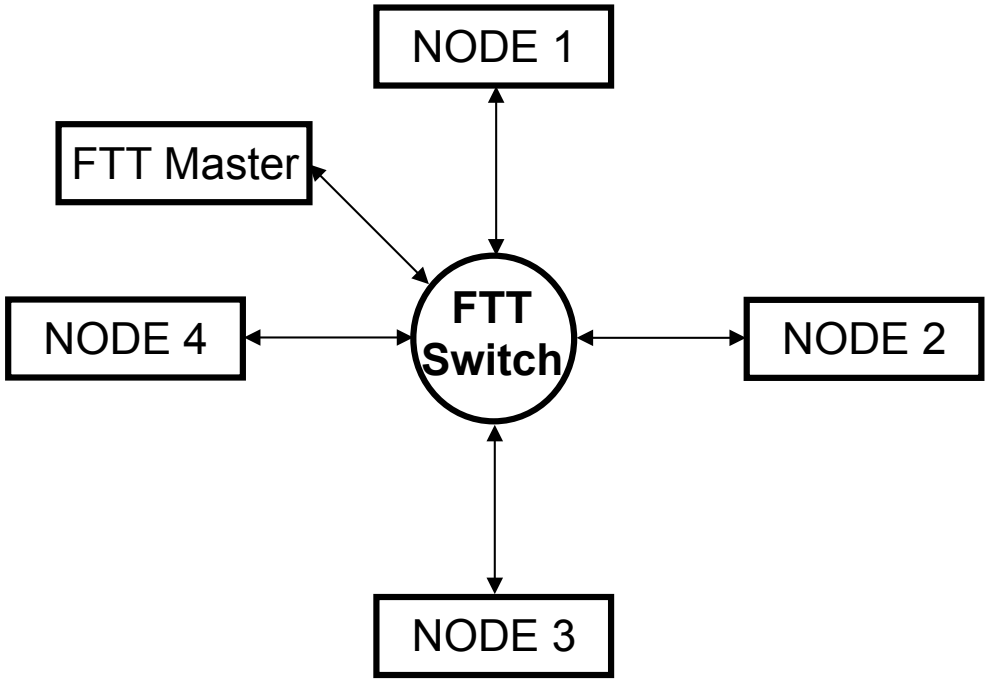
Our simplification (first step)

- **Our mechanisms are designed for the Synch-W,**
  - which is the one that provides more advantages in terms of FT



Building the Fault-Tolerant System

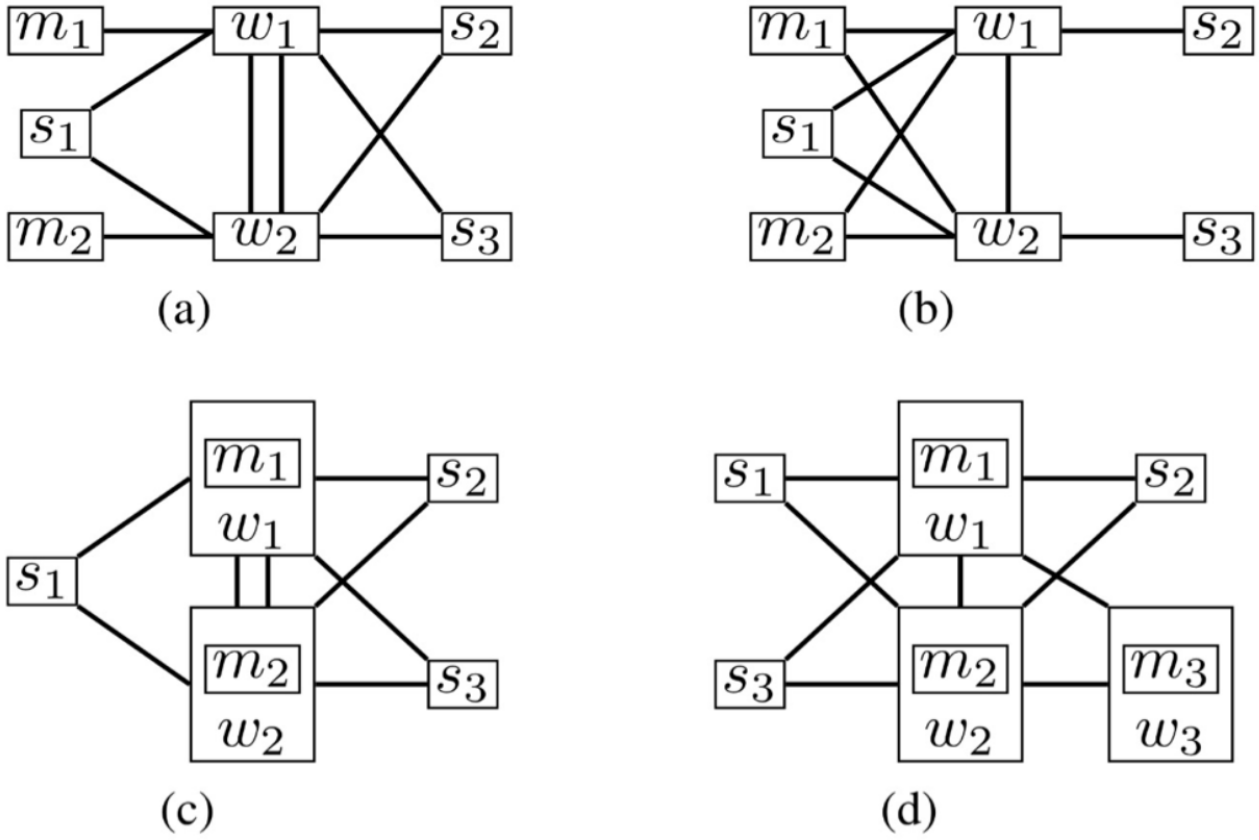
- Using HaRTES as switch was in the initial proposal
  BUT we try to see the problem with a wider perspective
  - compare the dependability of **different replicated architectures**



Building the Fault-Tolerant System

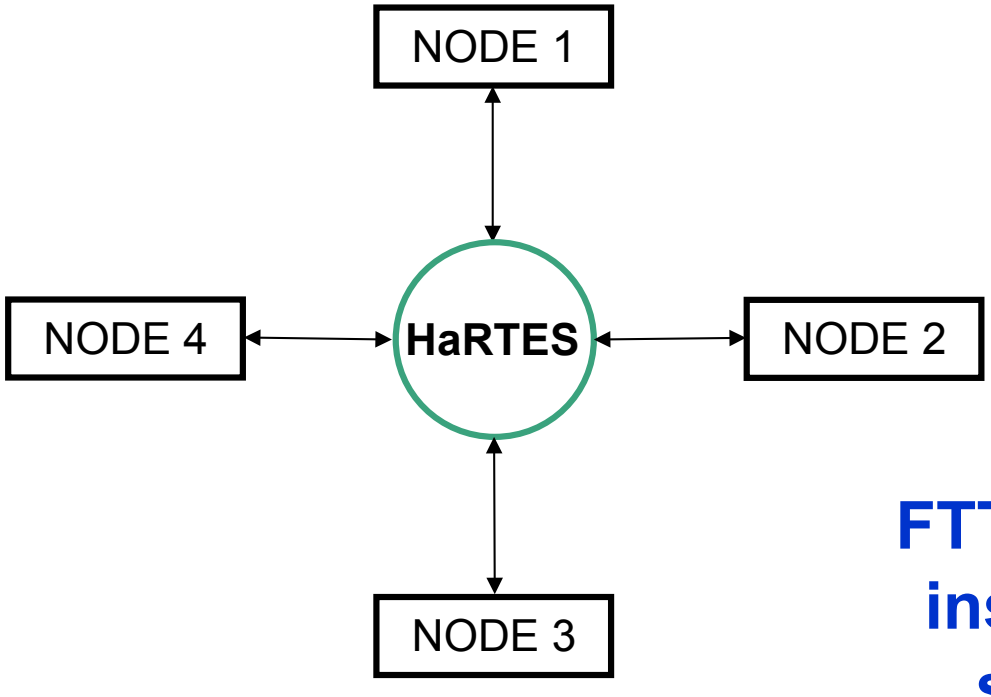
- Using HaRTES as switch was in the initial proposal
  BUT we try to see the problem with a wider perspective
  - compare the dependability of **different replicated architectures**

Generate the complete design space (and evaluate the reliability of each option)



Building the Fault-Tolerant System

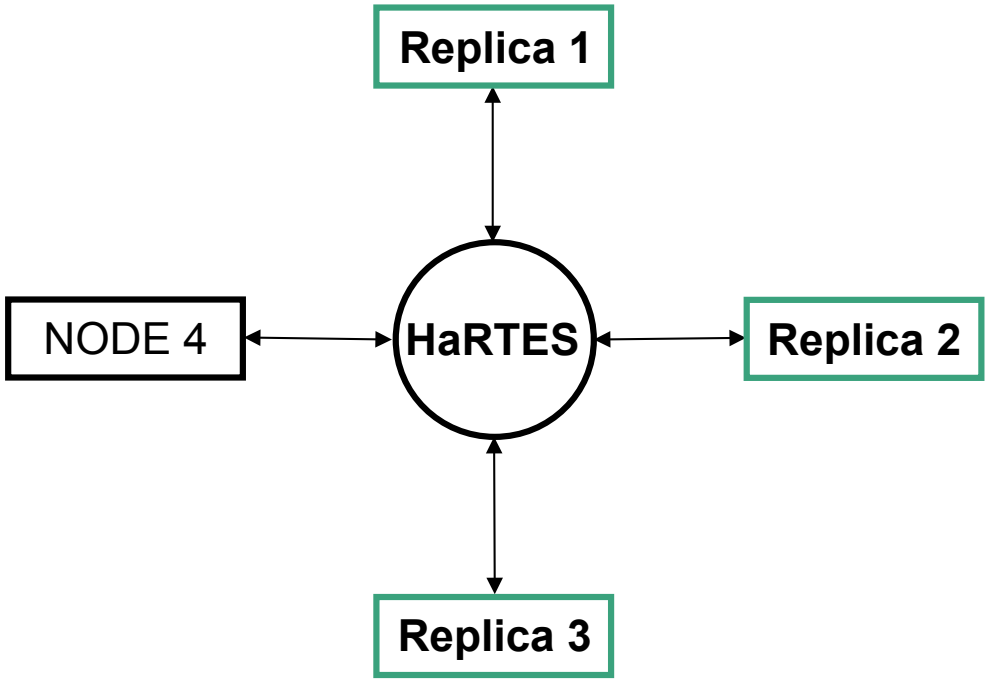
- For practical reasons we decided to retake HaRTES as corner-stone



FTT master inside the switch

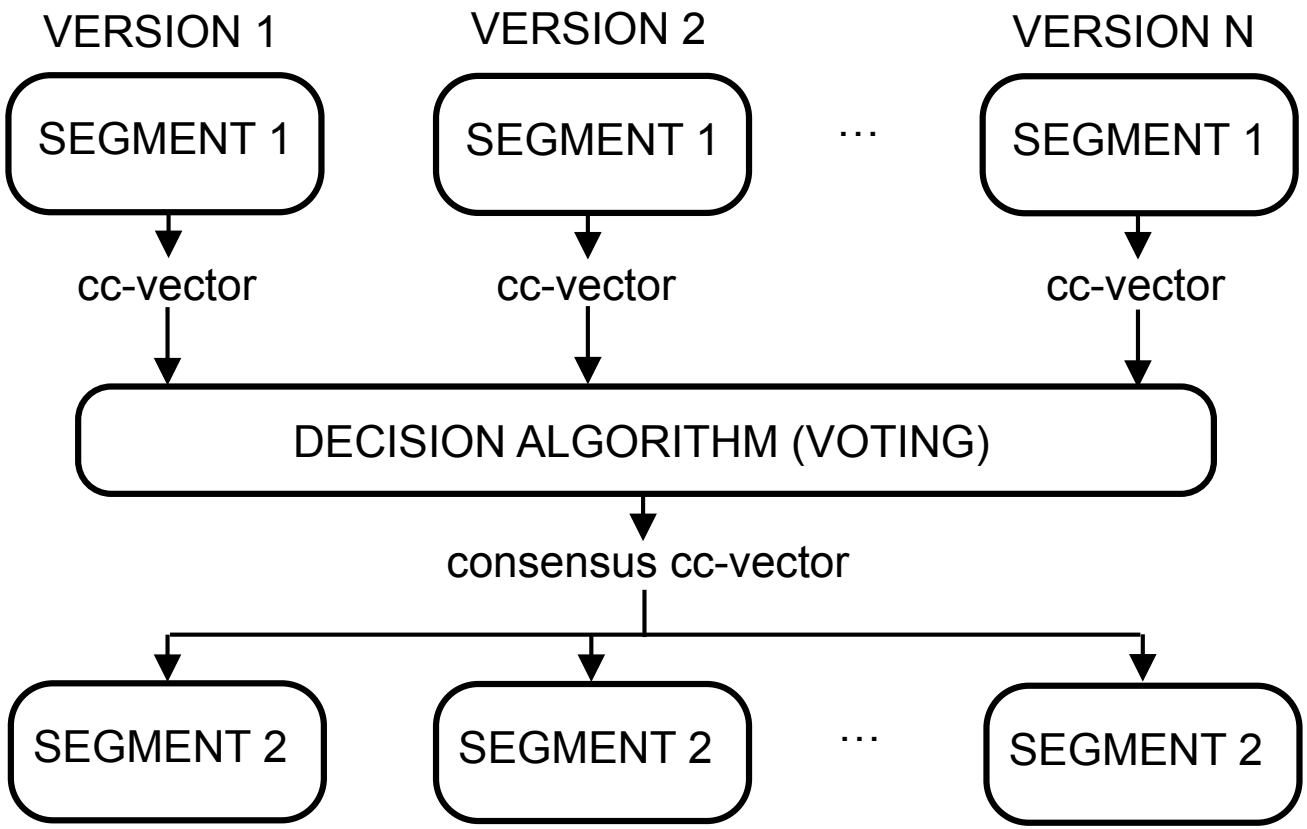
Node (FTT slave) Replication

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



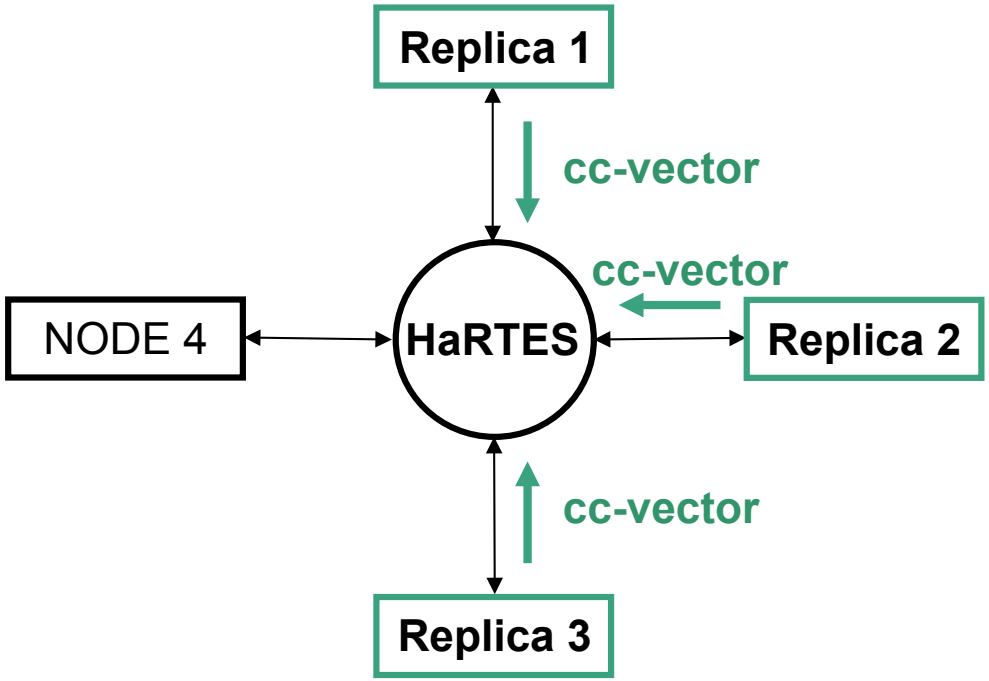
Node (FTT slave) Replication

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



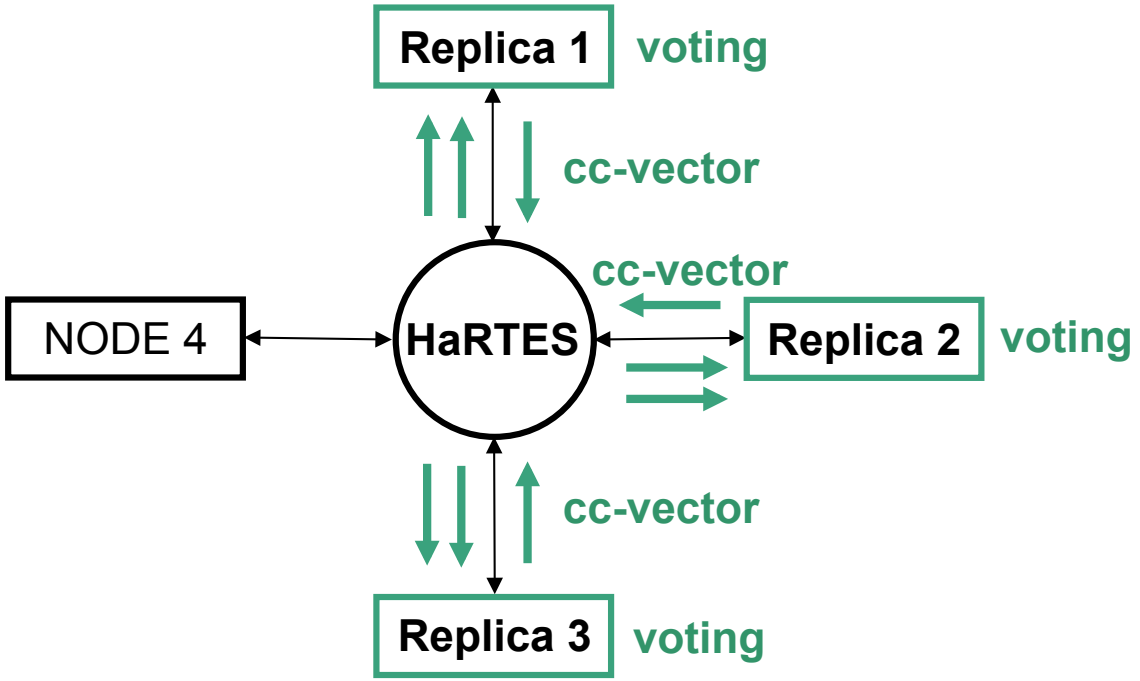
Node (FTT slave) Replication

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



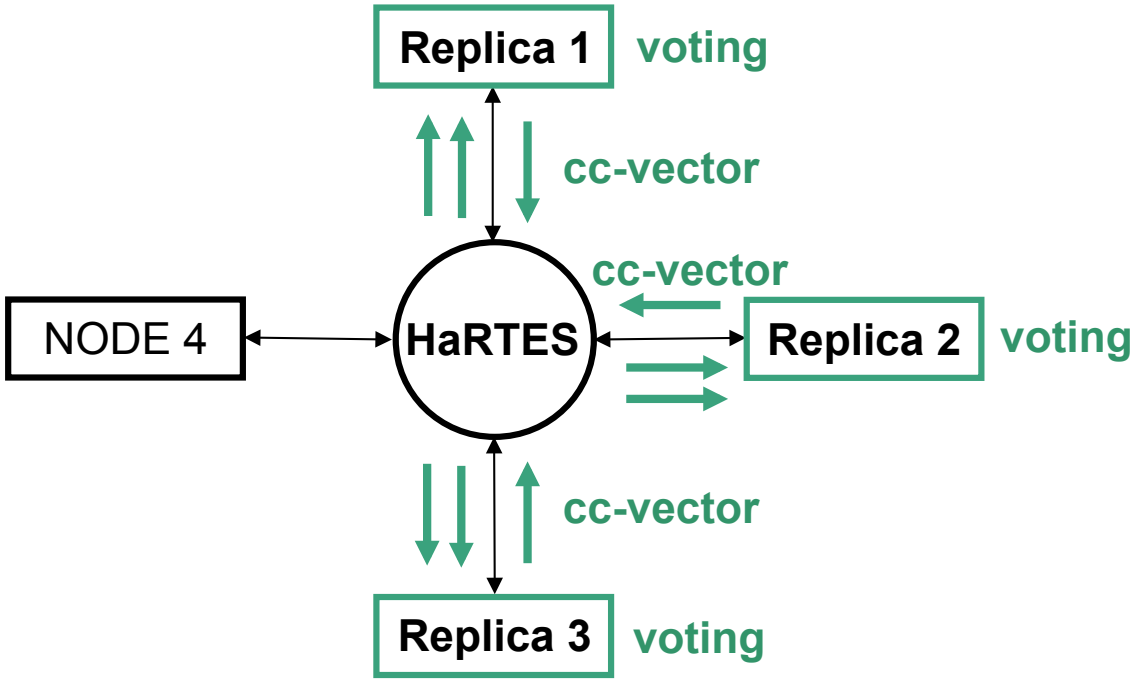
Node (FTT slave) Replication

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



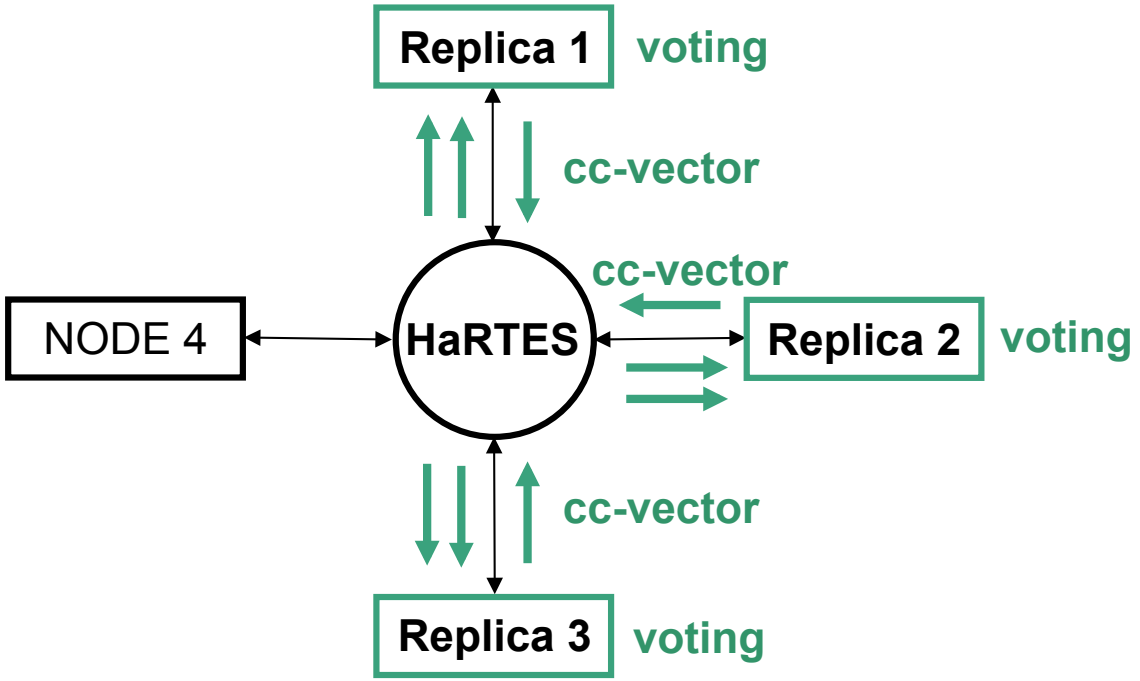
Node (FTT slave) Replication

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



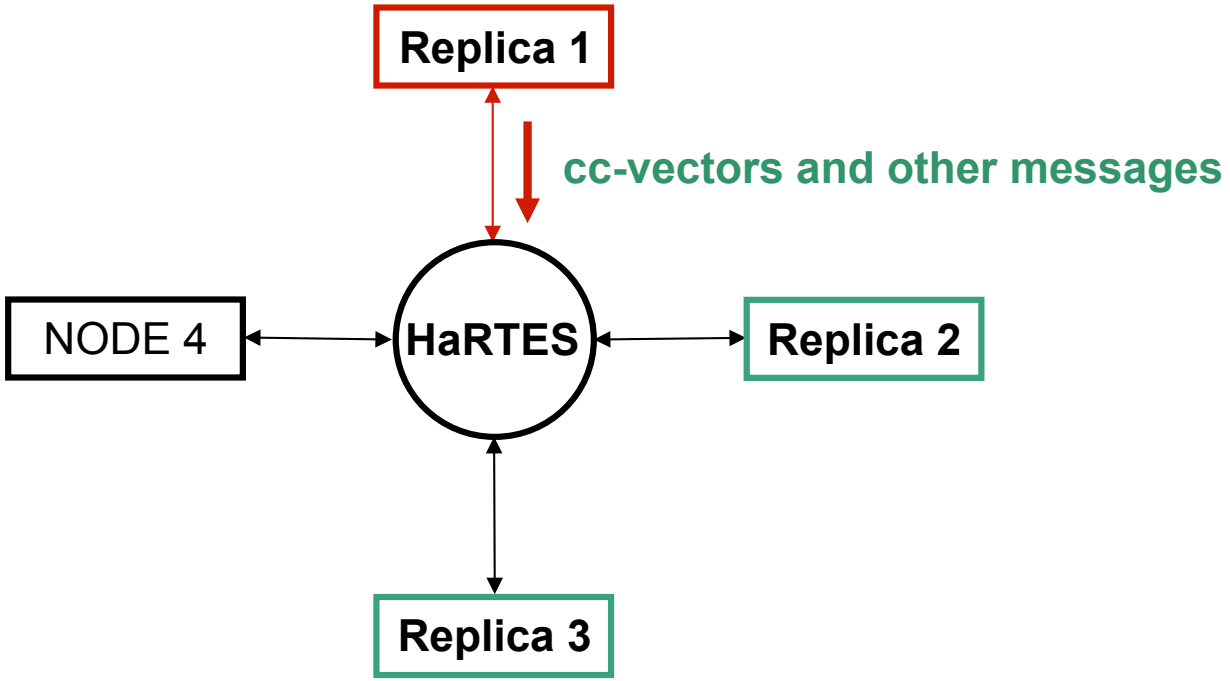
Node (FTT slave) Replication

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



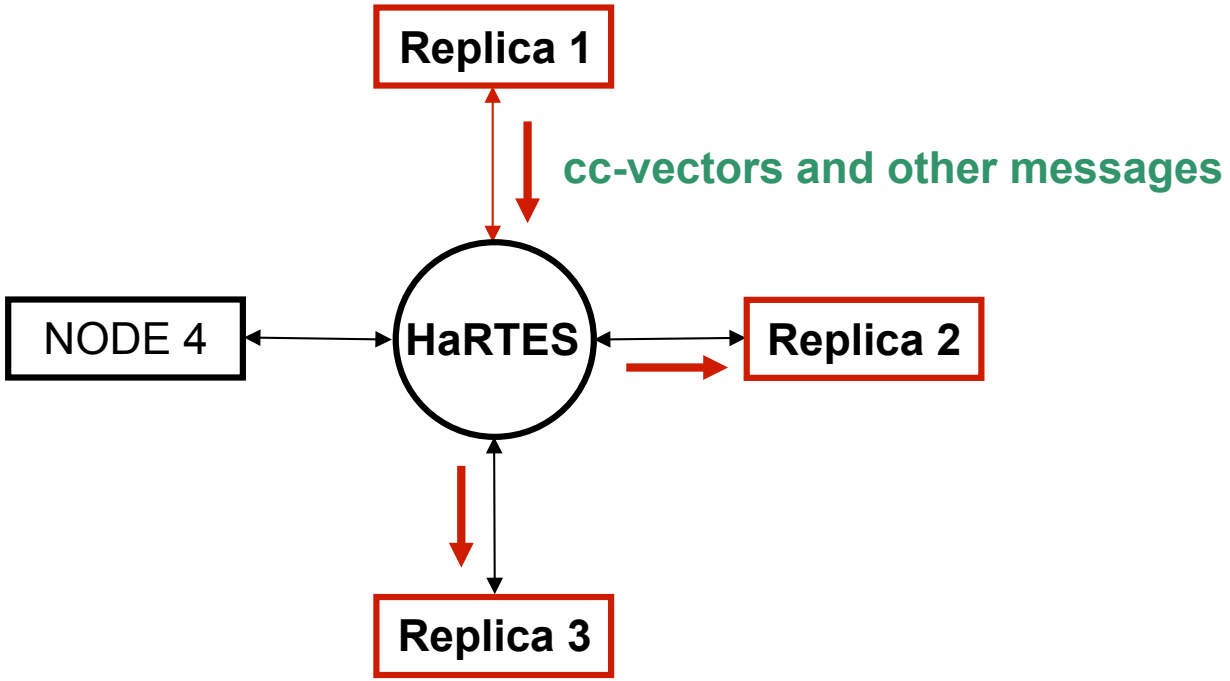
Independence of Failures

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



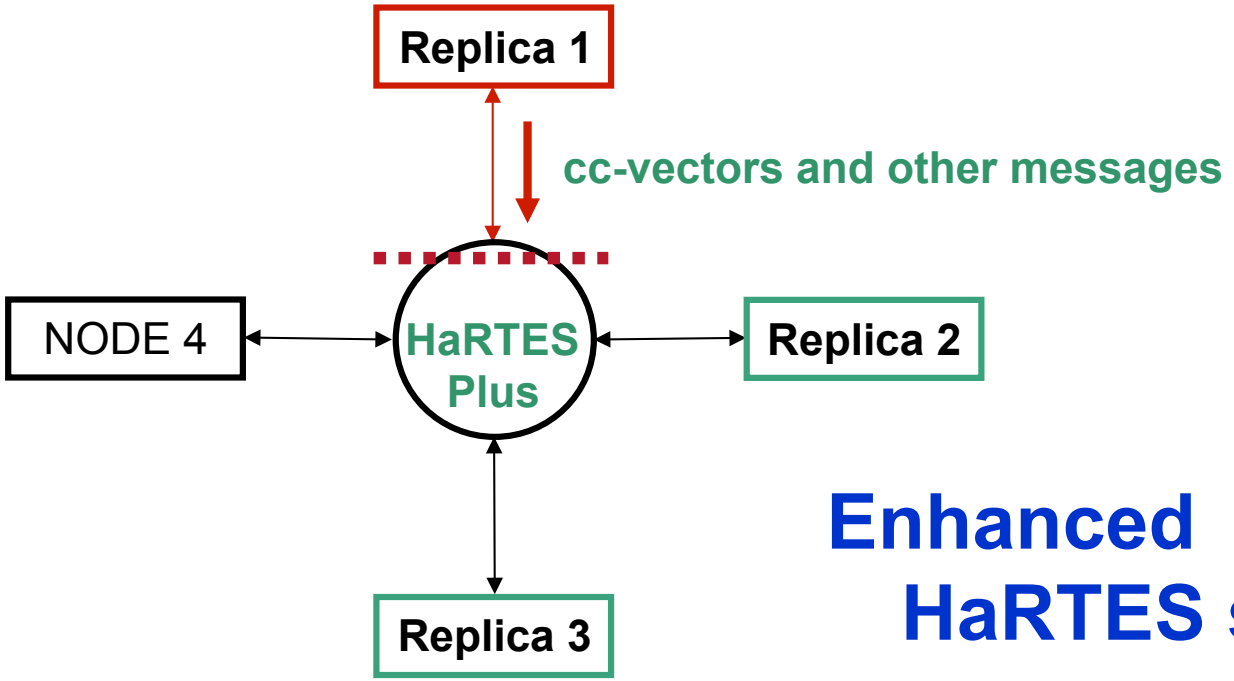
Independence of Failures

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



Independence of Failures

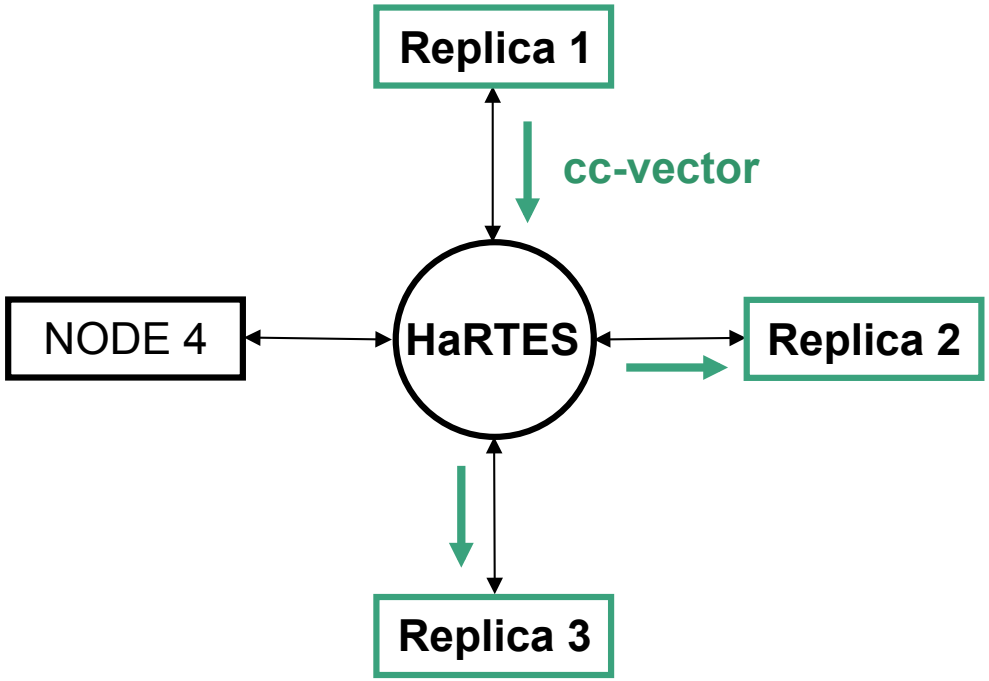
- **Two aspects**:
  - Replicas in different nodes, thus initially they are independent
  - However, a faulty replica or link can propagate errors. **Prevent it!!**



Enhanced HaRTES switch (port guardians)

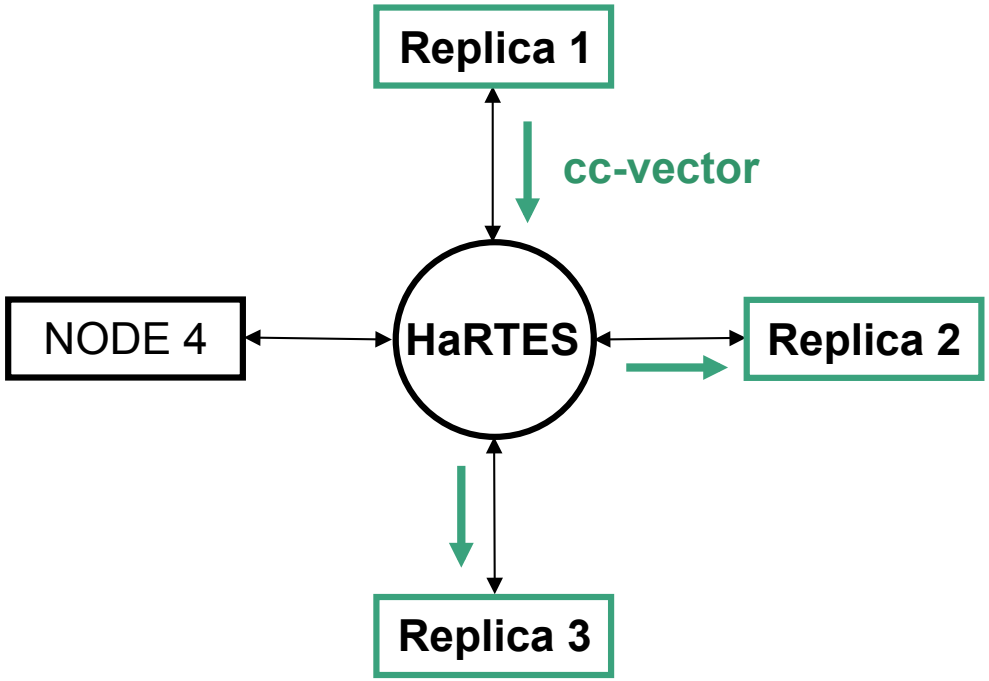
Consistent Communication

- Receiving the same cc-vectors helps consistent voting
  - Design a (TOB) **protocol that adapts/takes advantage of FTT**
  - Easier to achieve with **restricted failure semantics** for the replicas
  - Preventing **error propagation helps restricting** failure semantics



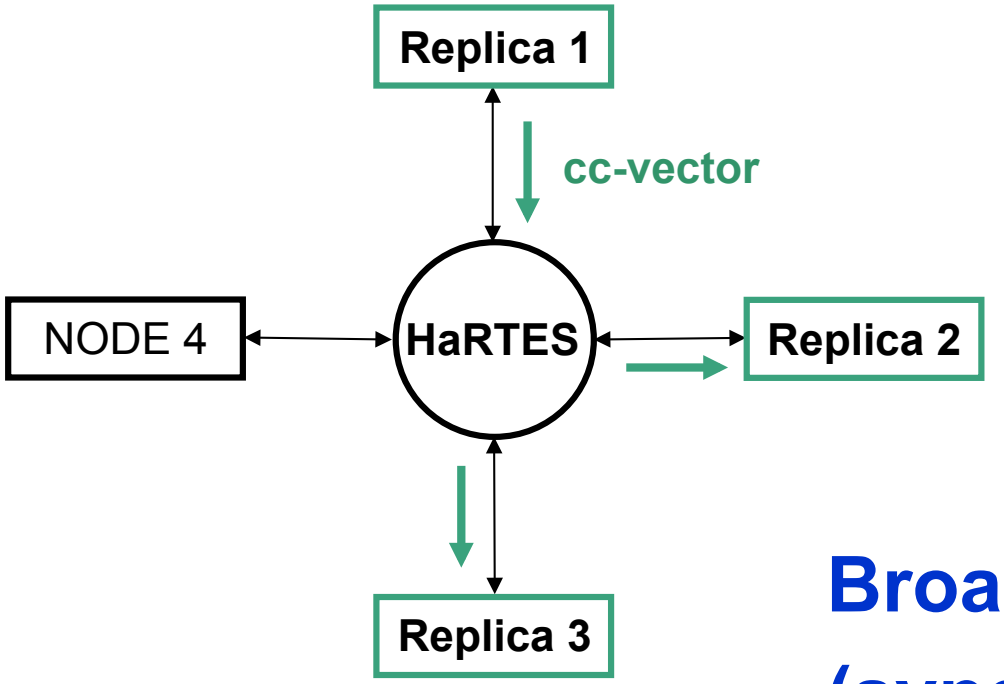
Consistent Communication

- A Total Order Broadcast Protocol for FTT: **TOPS**
  - Each receiver sends an ACK for each message
  - Central element collects them and sends a delivery indication



Consistent Communication

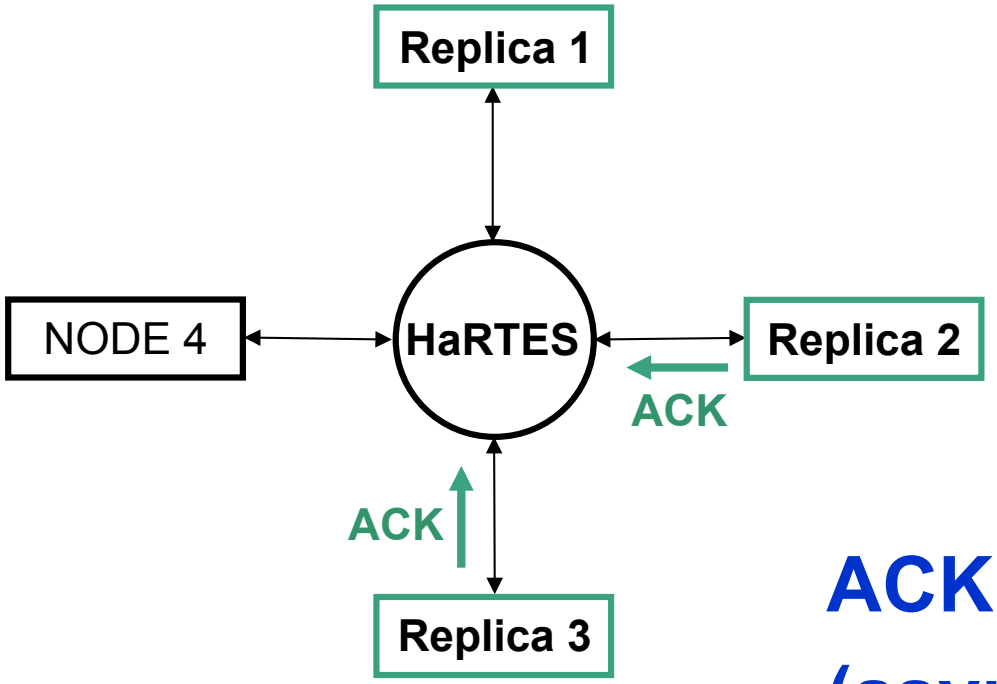
- A Total Order Broadcast Protocol for FTT: **TOPS**
  - Each receiver sends an ACK for each message
  - Central element collects them and sends a delivery indication



Broadcast (synch window)

Consistent Communication

- A Total Order Broadcast Protocol for FTT: **TOPS**
  - Each receiver sends an ACK for each message
  - Central element collects them and sends a delivery indication

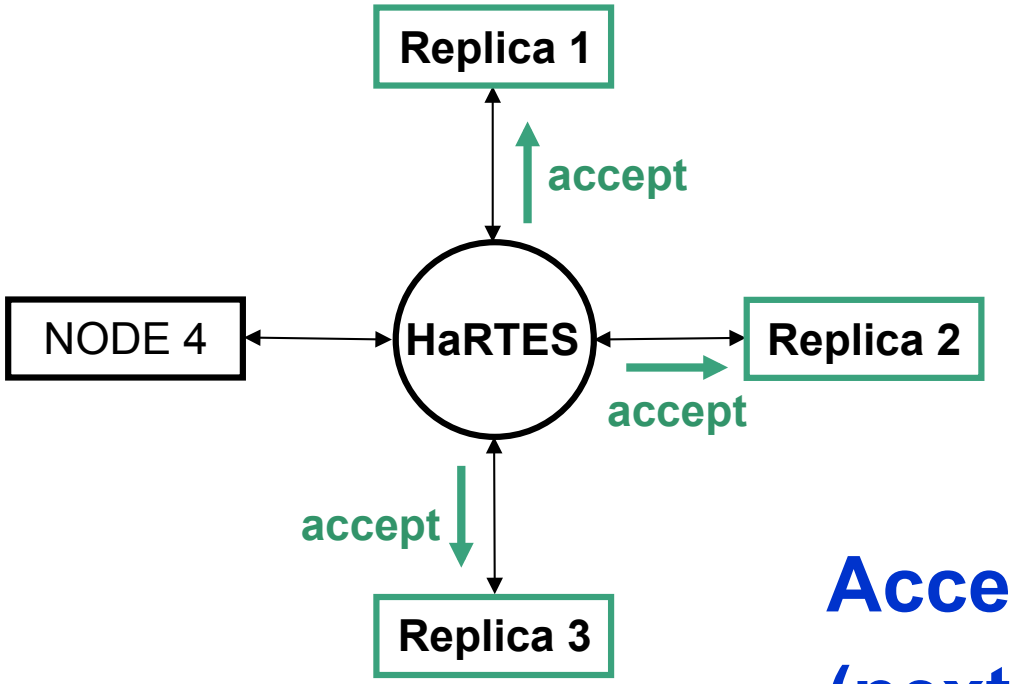


ACK

(asynch window)

Consistent Communication

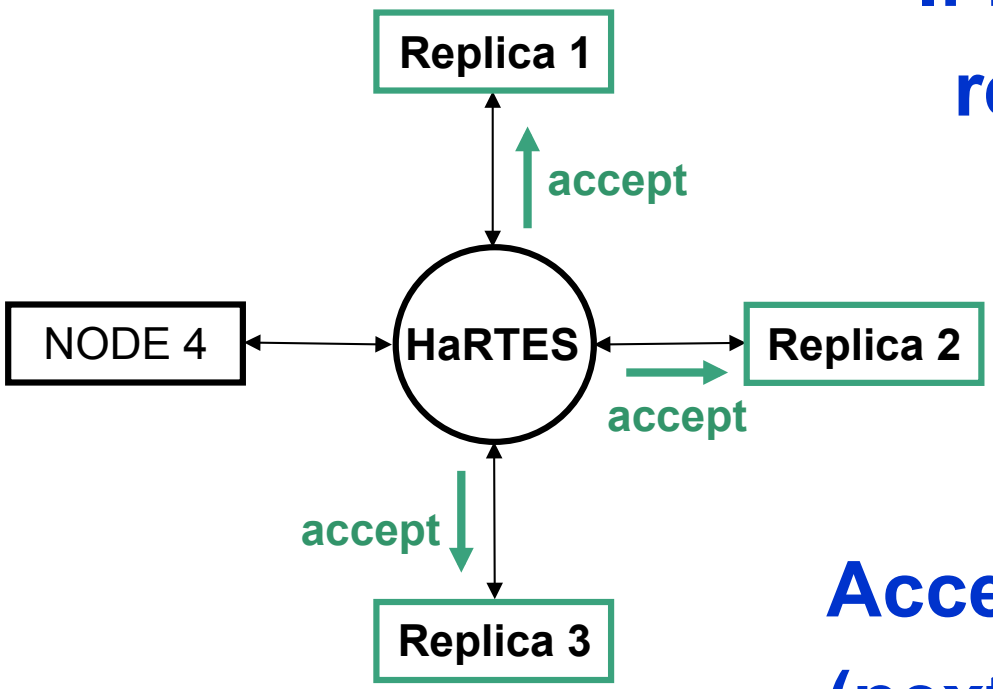
- A Total Order Broadcast Protocol for FTT: **TOPS**
  - Each receiver sends an ACK for each message
  - Central element collects them and sends a delivery indication



Accept indication (next TM)

Consistent Communication

- A Total Order Broadcast Protocol for FTT: **TOPS**
  - Each receiver sends an ACK for each message
  - Central element collects them and sends a delivery indication

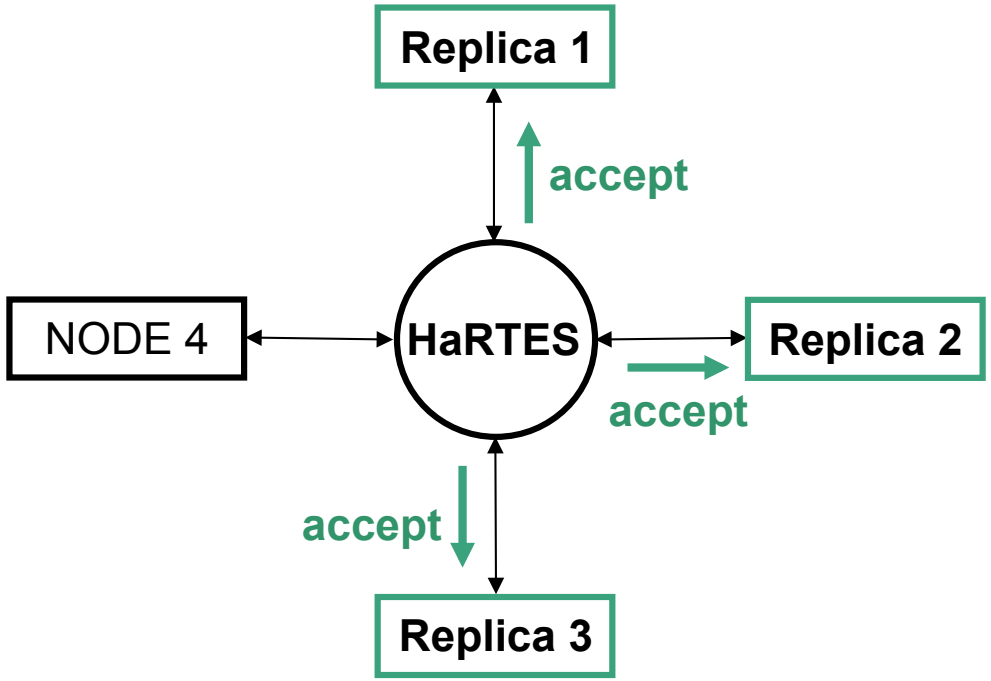


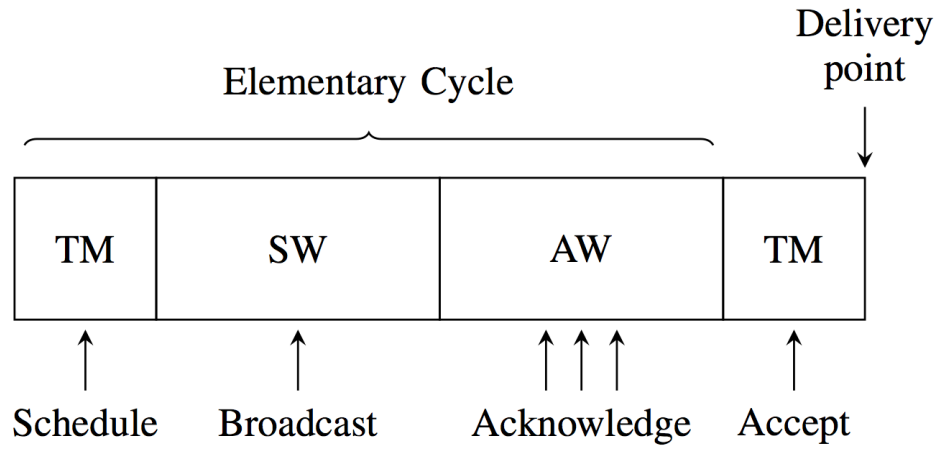
If not accept: reschedule

Accept indication (next TM)

Consistent Communication

- A Total Order Broadcast Protocol for FTT: **TOPS**
  - Each receiver sends an ACK for each message
  - Central element collects them and sends a delivery indication



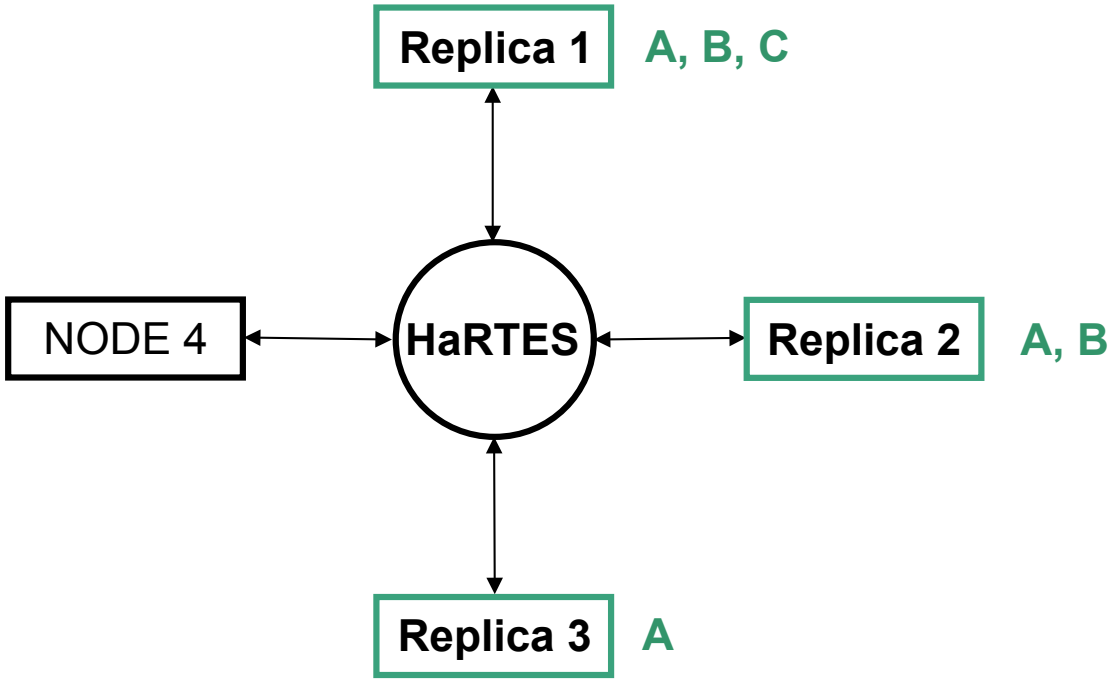


Distribution of the 4 phases over the EC structure.

Msgs and Acks sent redundantly

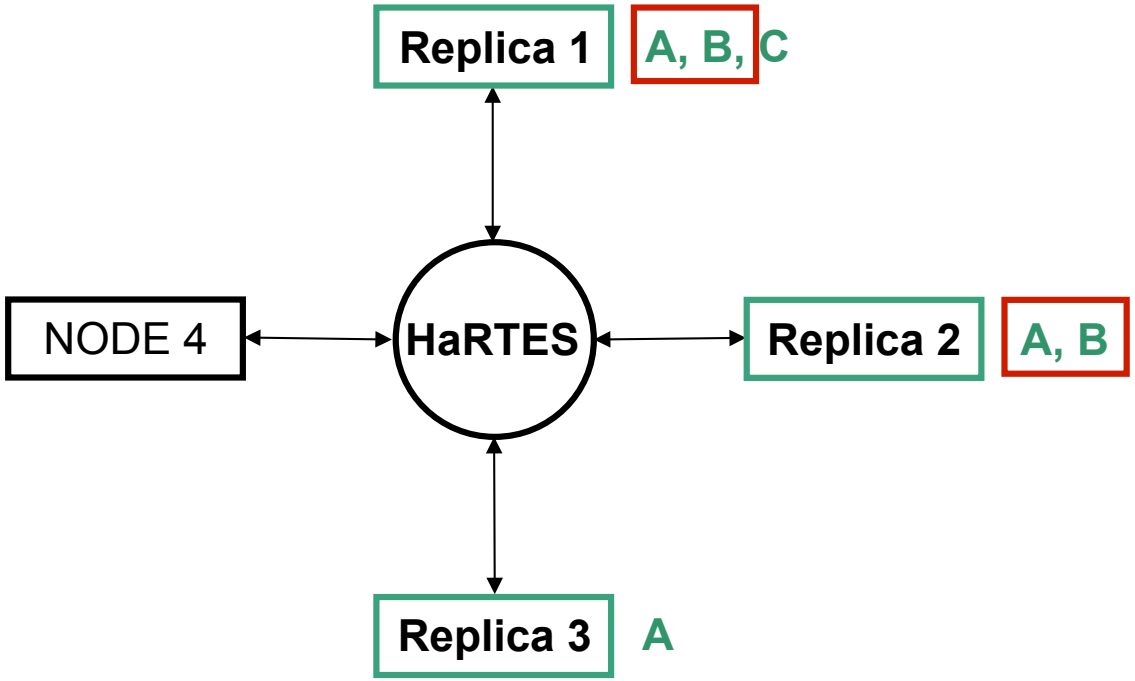
Consistent Communication

- **Adapting TOPS** for replicated voting: **CVEP-VSUA**
  - TOPS addresses each exchanged message (cc-vector)
  - Each replica could have a different set of messages
  - **Decide in which replicas to vote and with which messages**



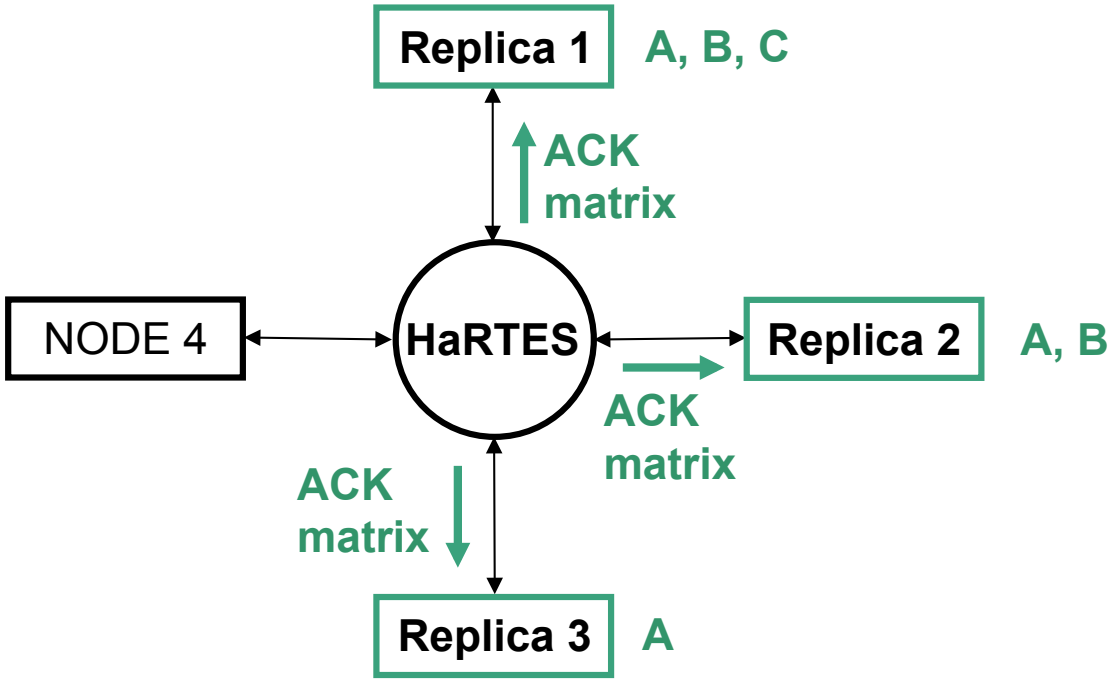
Consistent Communication

- **Adapting TOPS** for replicated voting: **CVEP-VSUA**
  - TOPS addresses each exchanged message (cc-vector)
  - Each replica could have a different set of messages
  - **Decide in which replicas to vote and with which messages**



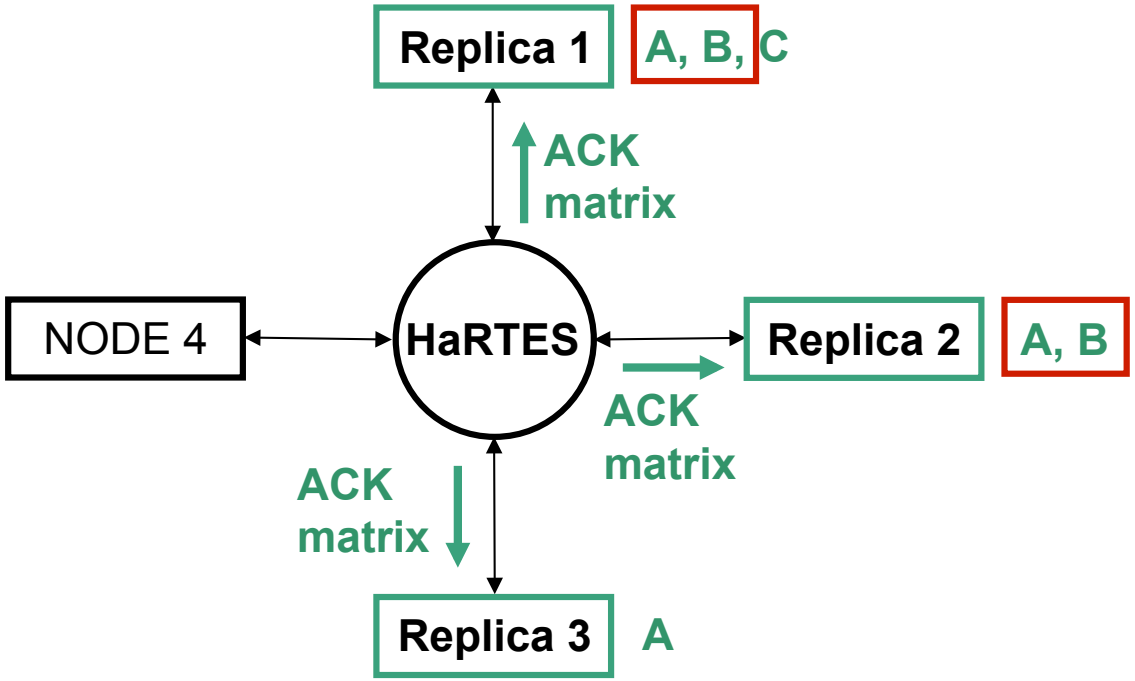
Consistent Communication

- **CVEP (Cc-Vector Exchange Protocol)** – VSUA
  - Increases reliability by relying in HaRTES for msg retransmission
  - Also HaRTES gathers all ACKs in a matrix
  - The matrix is sent to the nodes in the next TM for decision



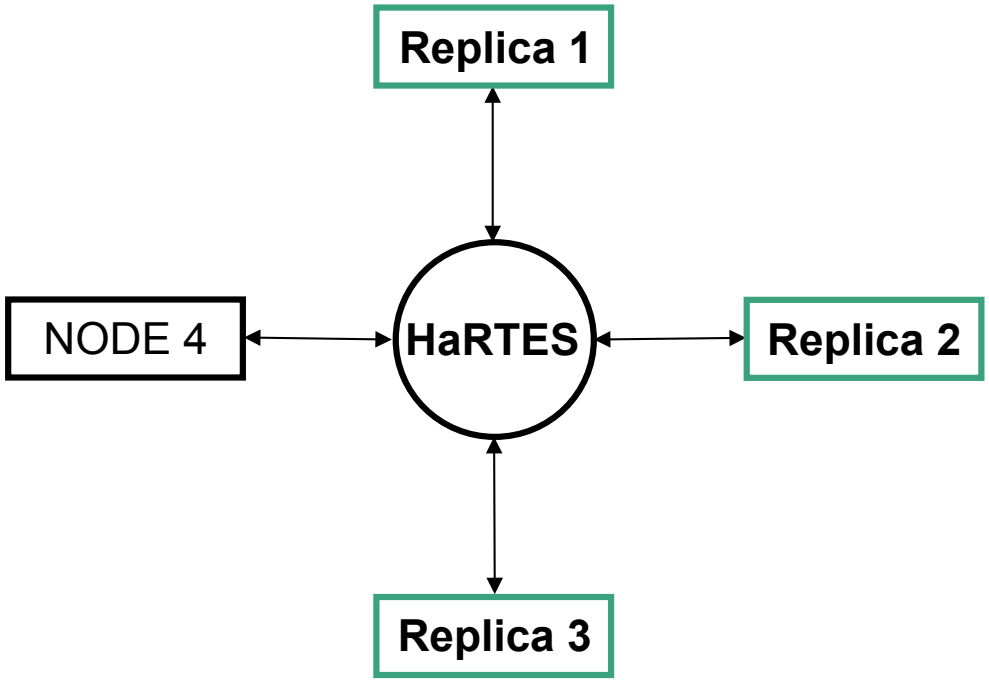
Consistent Communication

- CVEP – **VSUA (Voting Set Up Algorithm)**
  - Provides a kind of best-effort interactive consistency
  - Works as long as a majority of non faulty replicas consistently exchange a majority of messages



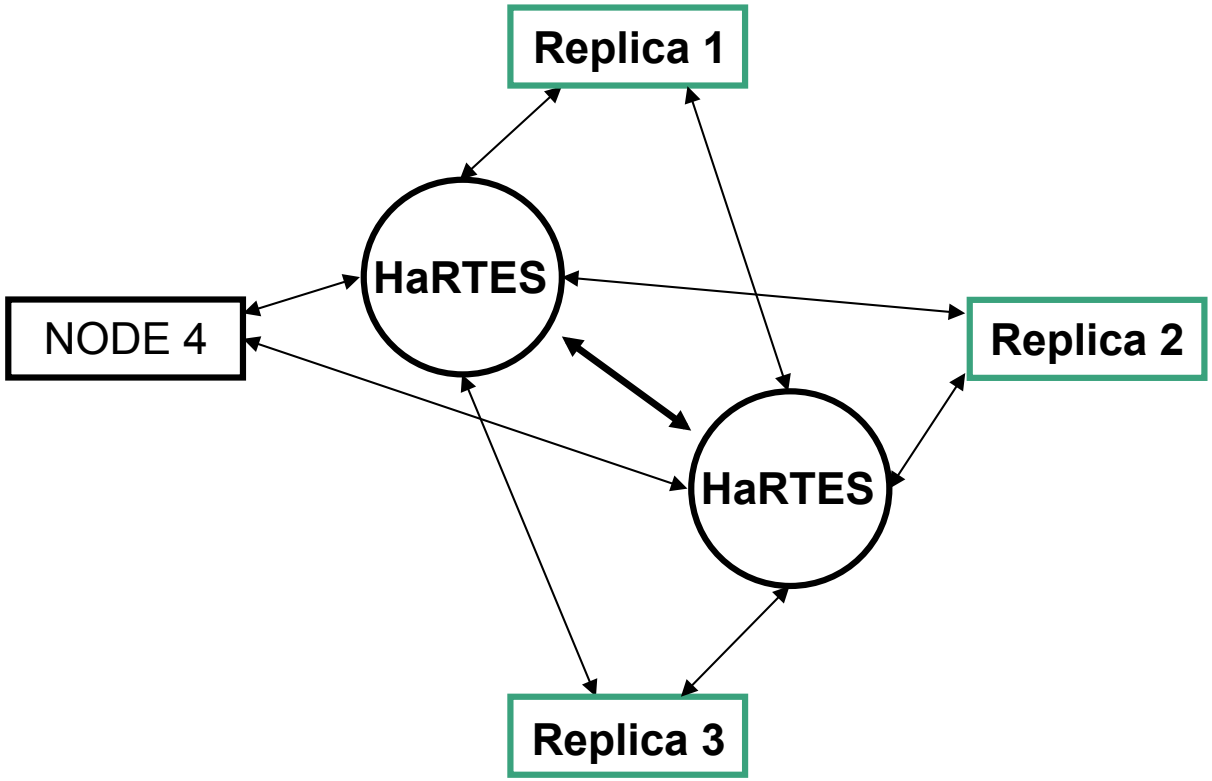
Channel Replication

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



Channel (Spatial) Replication

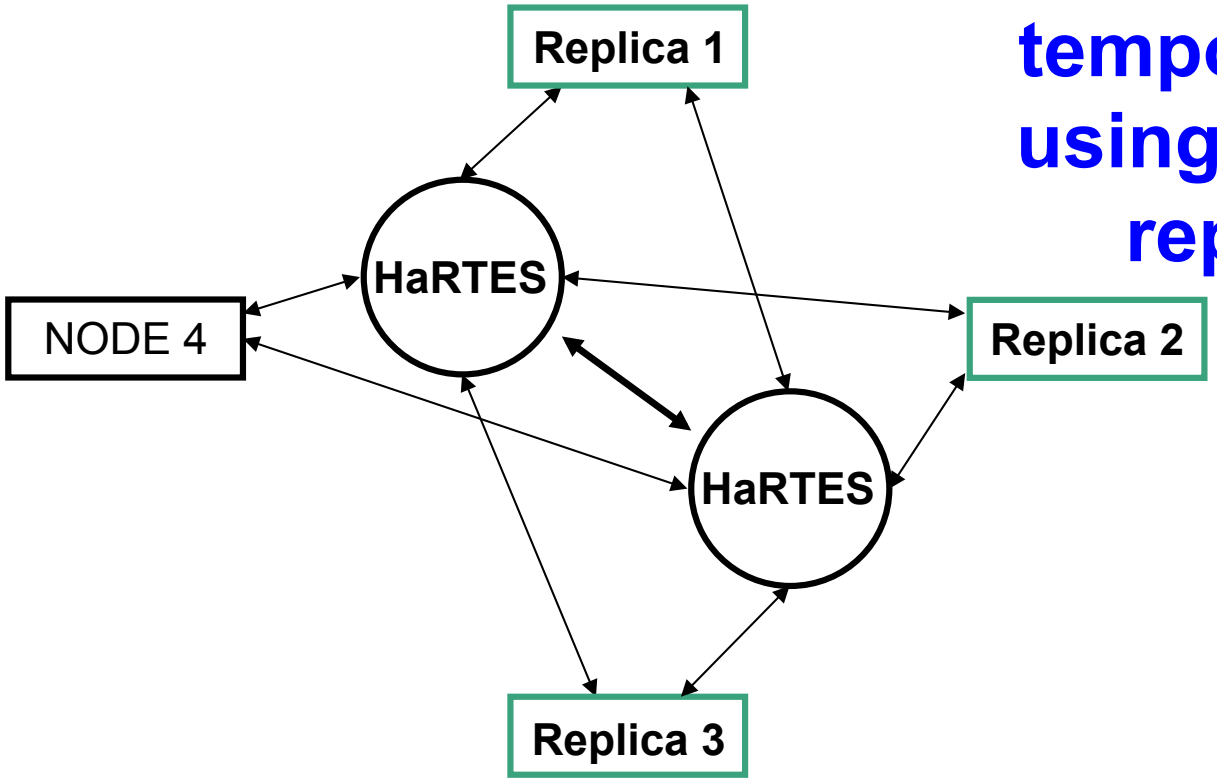
- Otherwise, **master and switch** are **single points of failure**
  - Therefore we replicate as shown below



Channel (Spatial) Replication

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

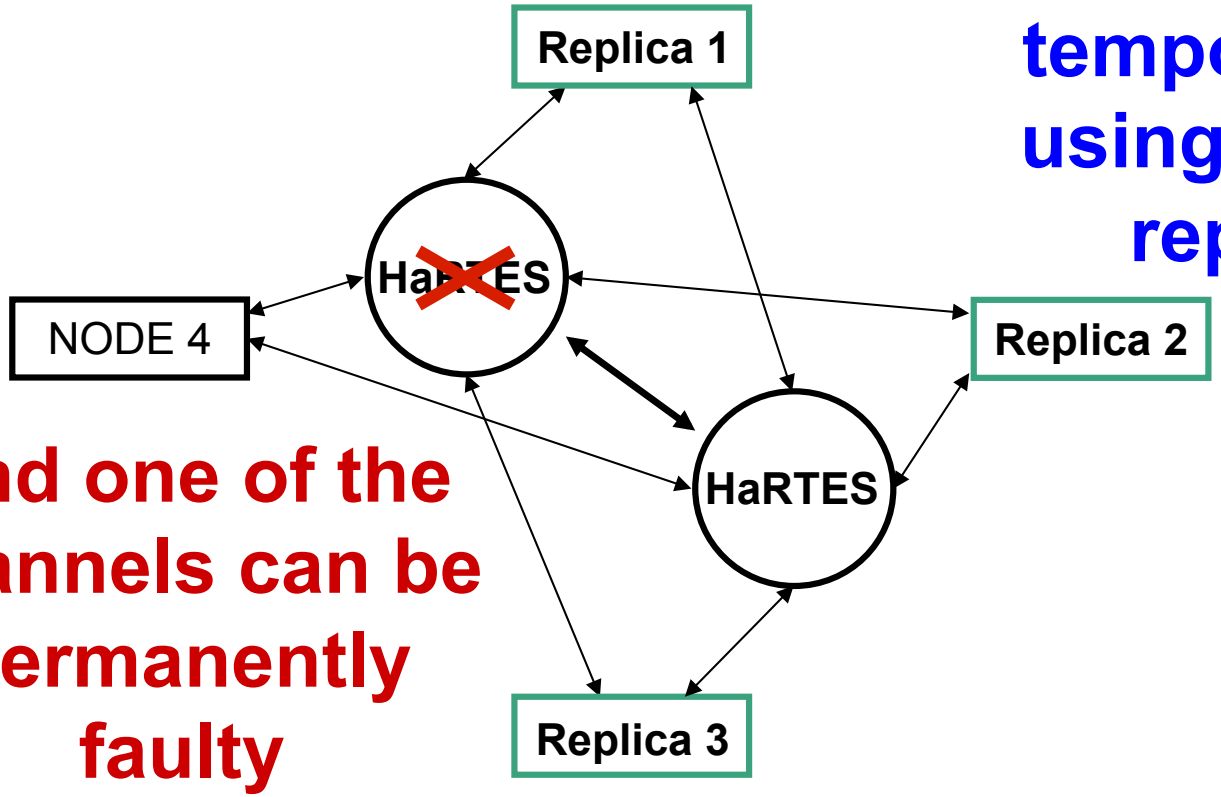
BUT it is a waste to tolerate temporary faults using the spatial replication



Channel (Spatial) Replication

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

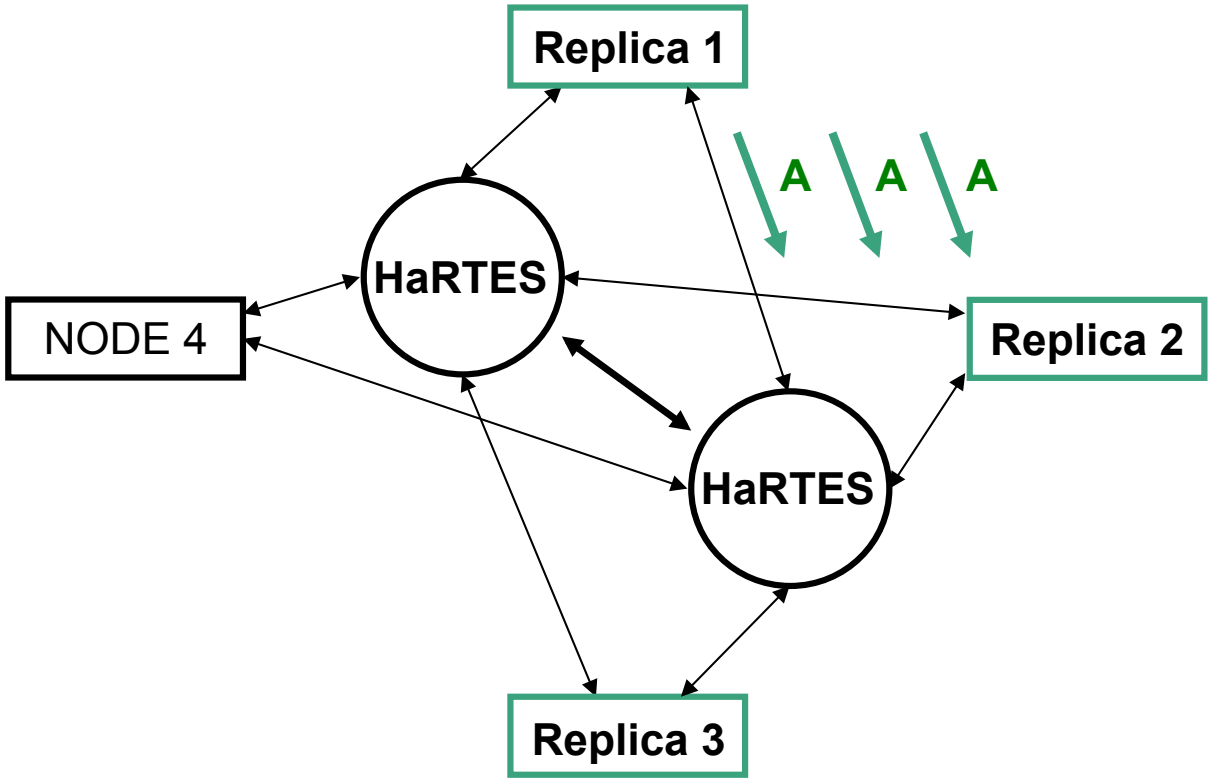
BUT it is a waste to tolerate temporary faults using the spatial replication



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

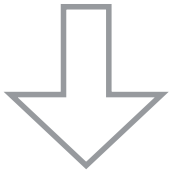
'Tis the lesson we shall heed
Try, try, try again
Just in case we don't succeed
Try, try, try again

  - **Predictable and deterministic**

Channel (Temporal) Replication

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

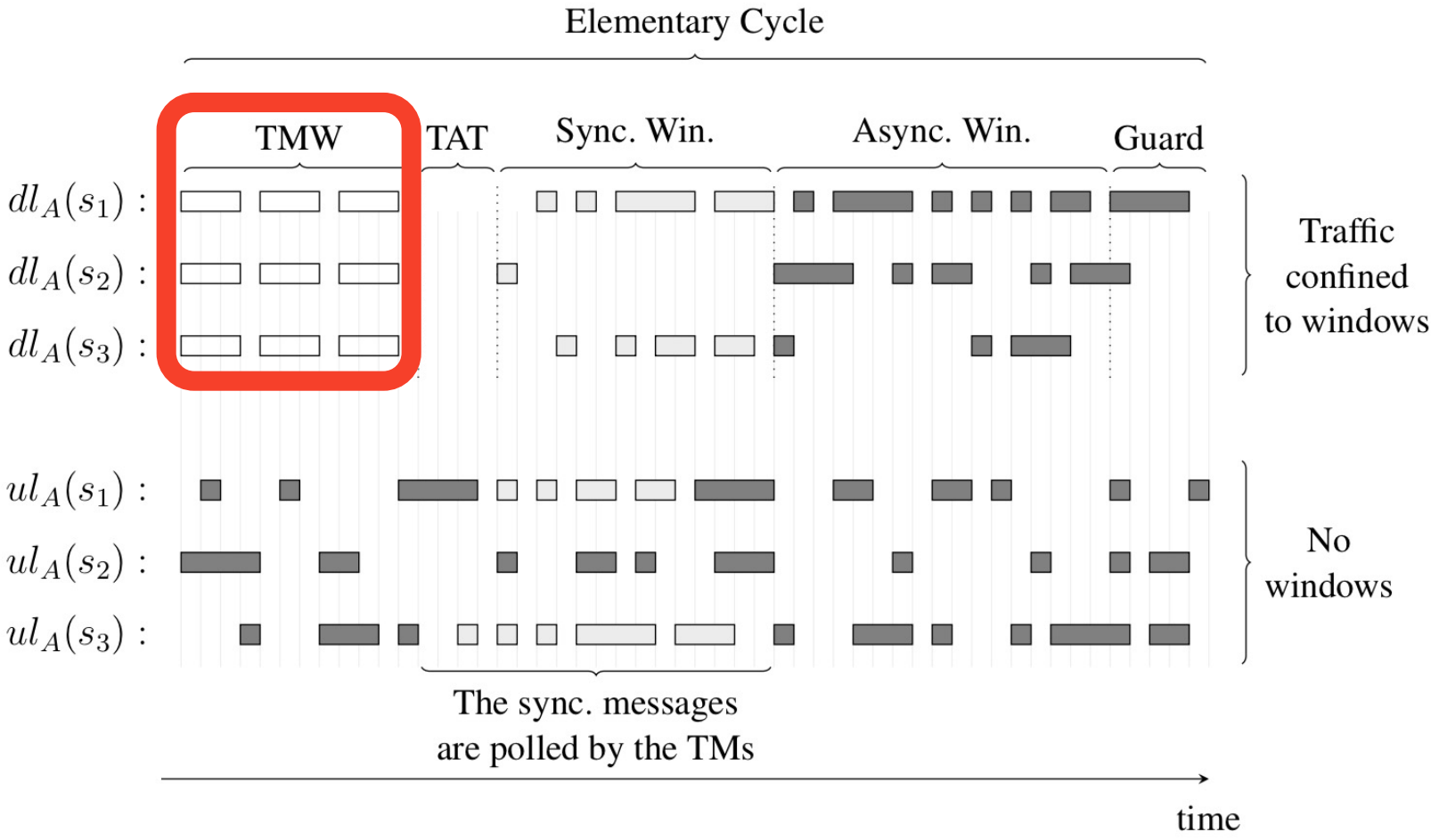
$$\mathrm{SRT} = \{m_i \mid m_i = (C_i, D_i, T_i, O_i, P_i), i \in [1, N_\mathrm{S}]\}$$
$$\mathrm{ART} = \{m_i \mid m_i = (C_i, D_i, I_i, P_i), i \in [1, N_\mathrm{A}]\}.$$



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

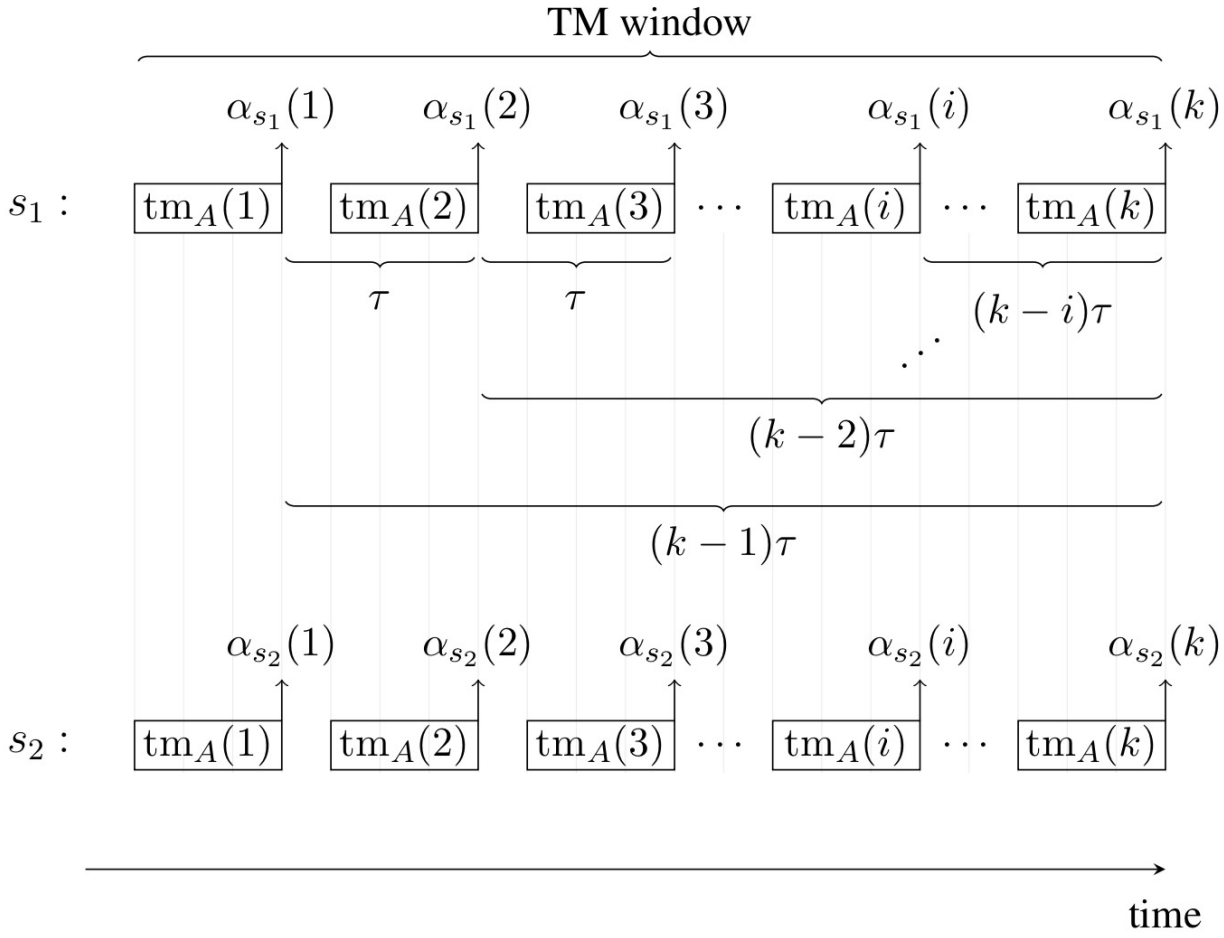
Channel (Temporal) Replication

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



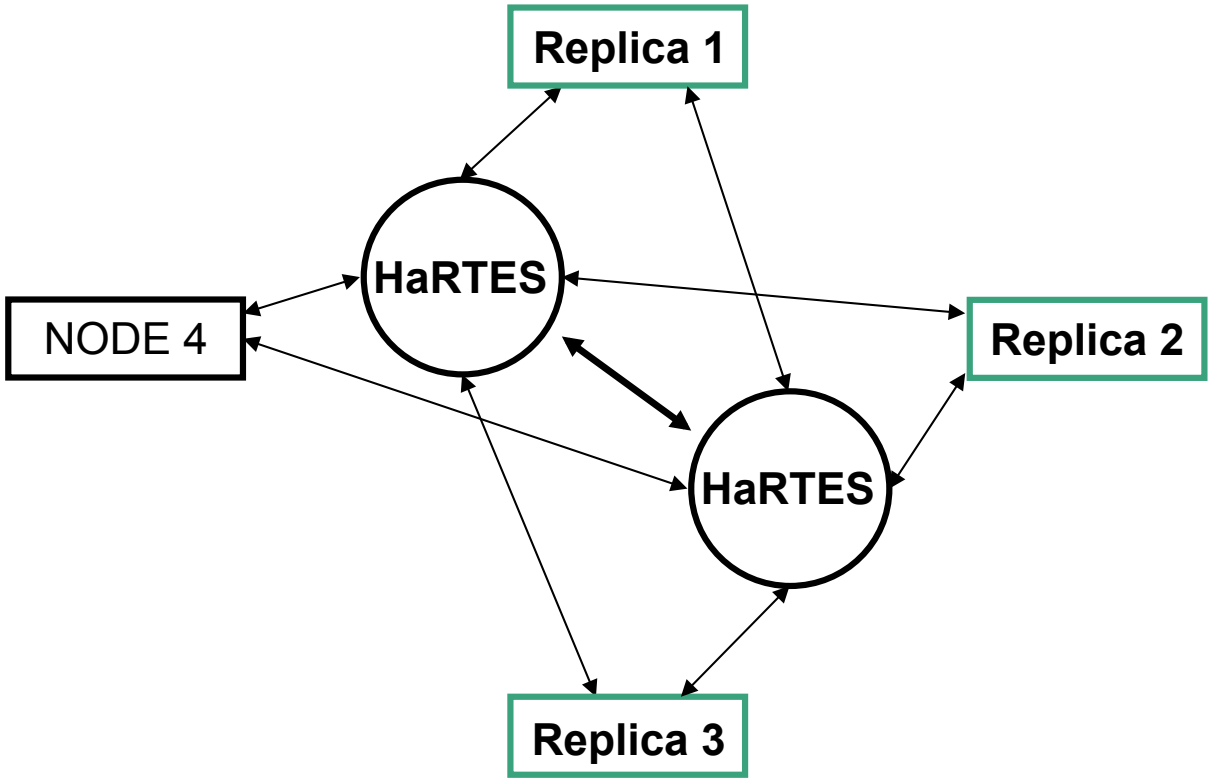
Channel (Temporal) Replication

- Therefore this changes the **EC synchronization w. slaves:**
  - Isochronous TM transmission



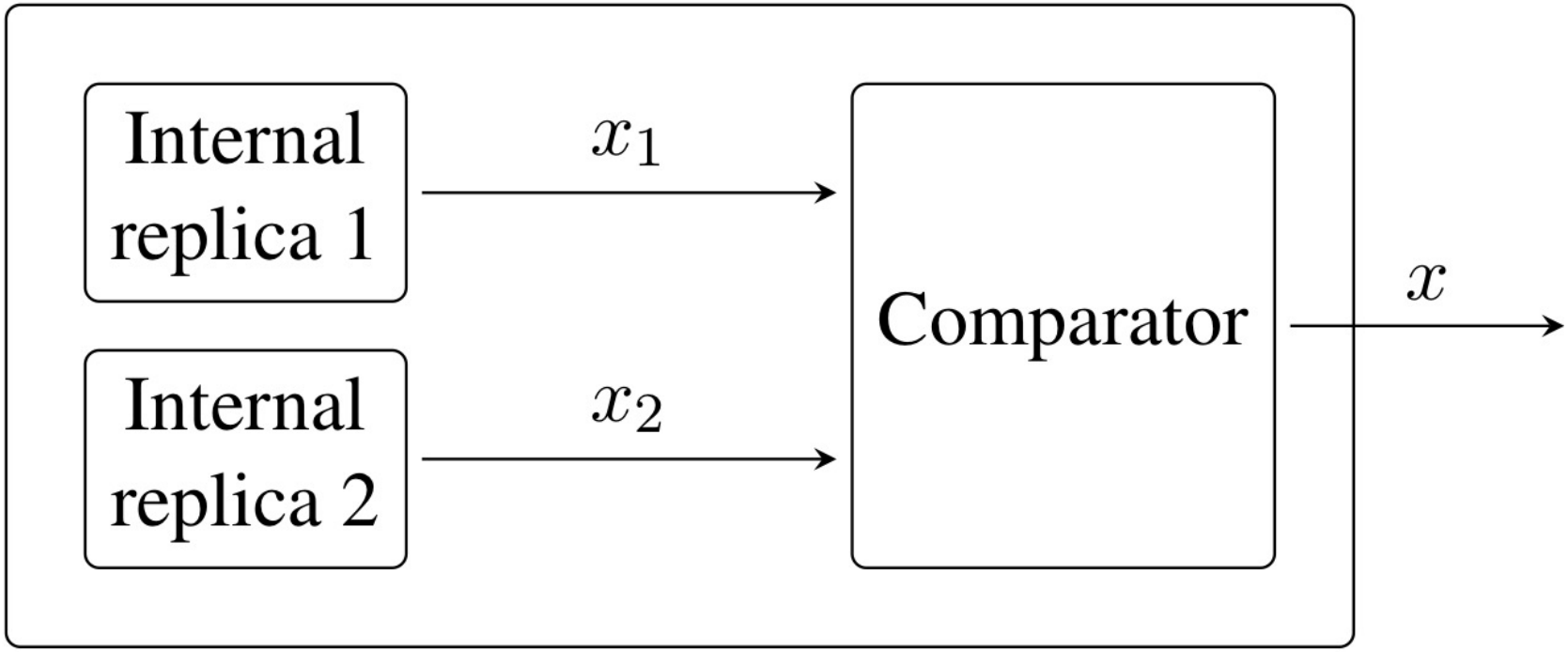
Channel (Spatial) Replication

- Back to spatial replication it is necessary to note that by **replicating the master 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**



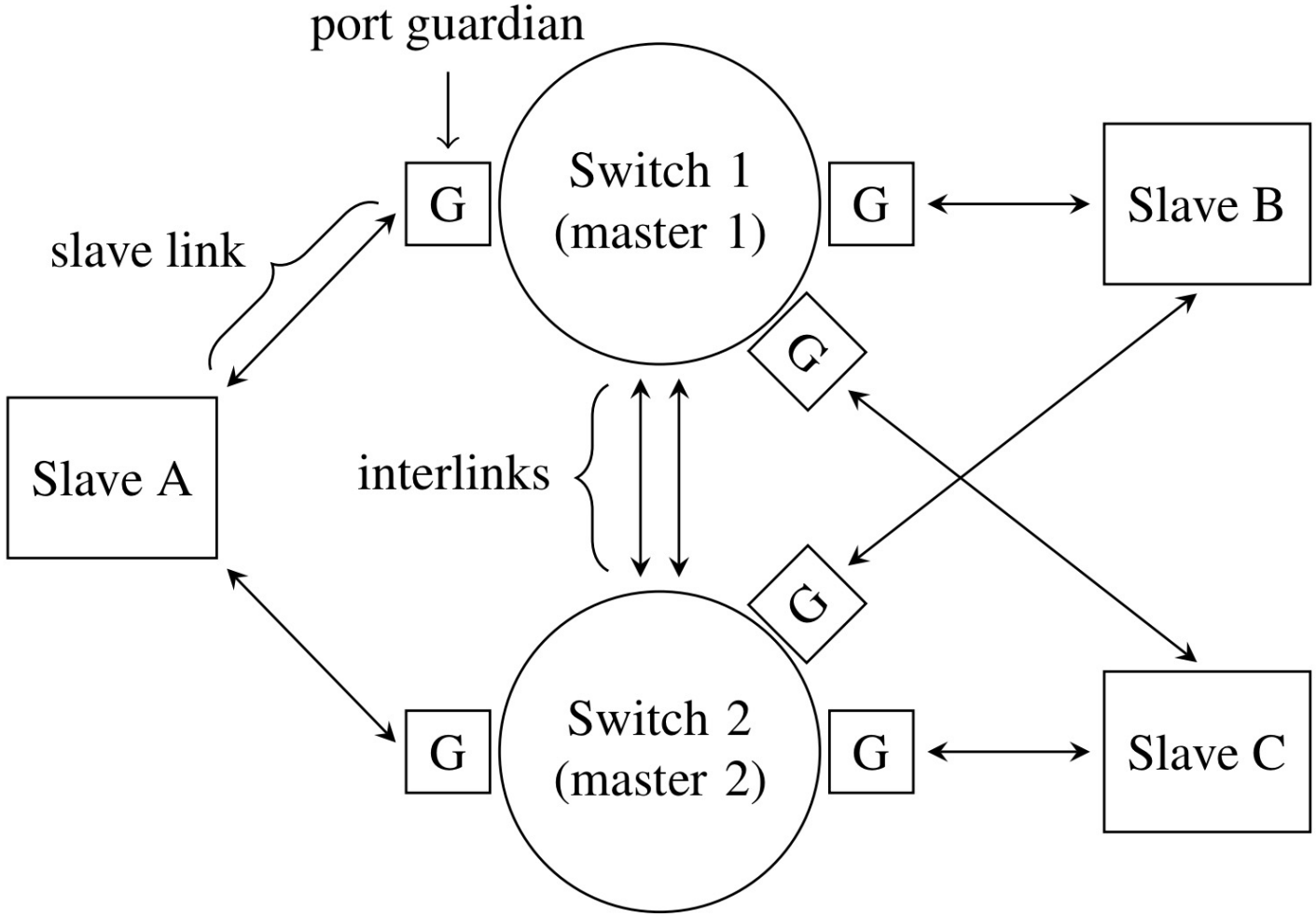
Channel (Spatial) Replication

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



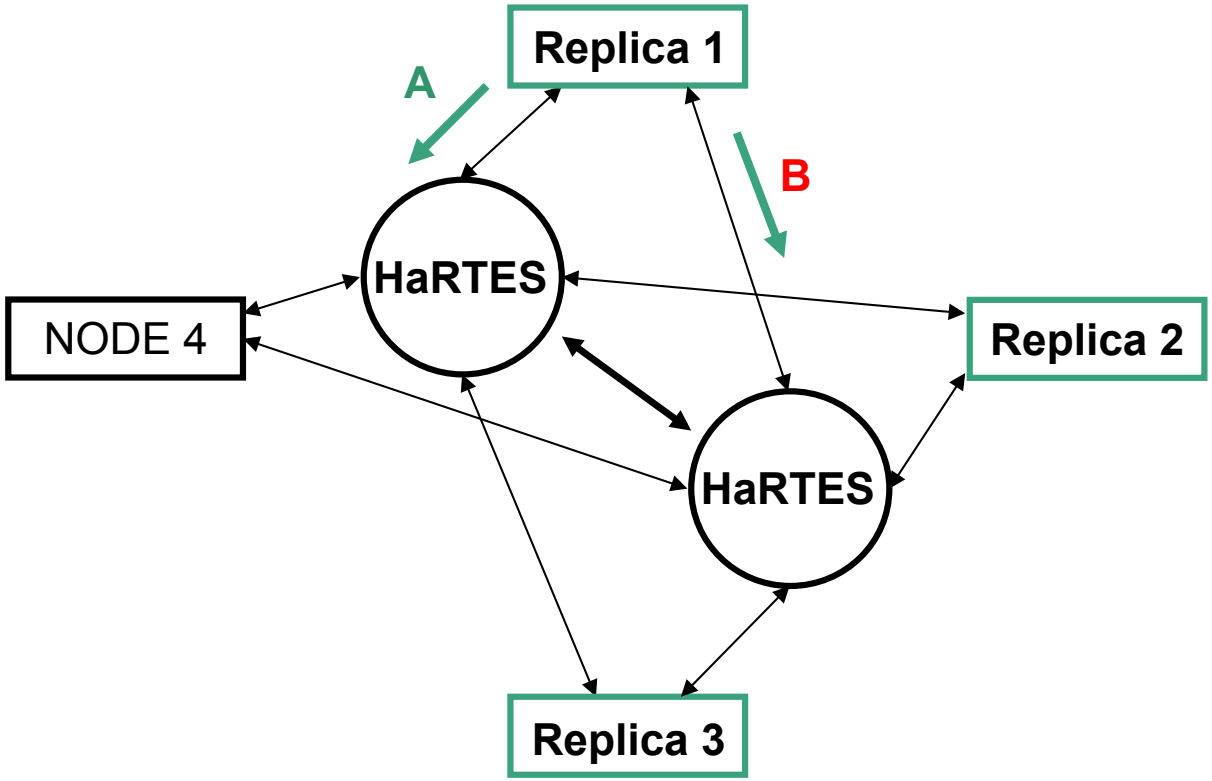
Channel (Spatial) Replication

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



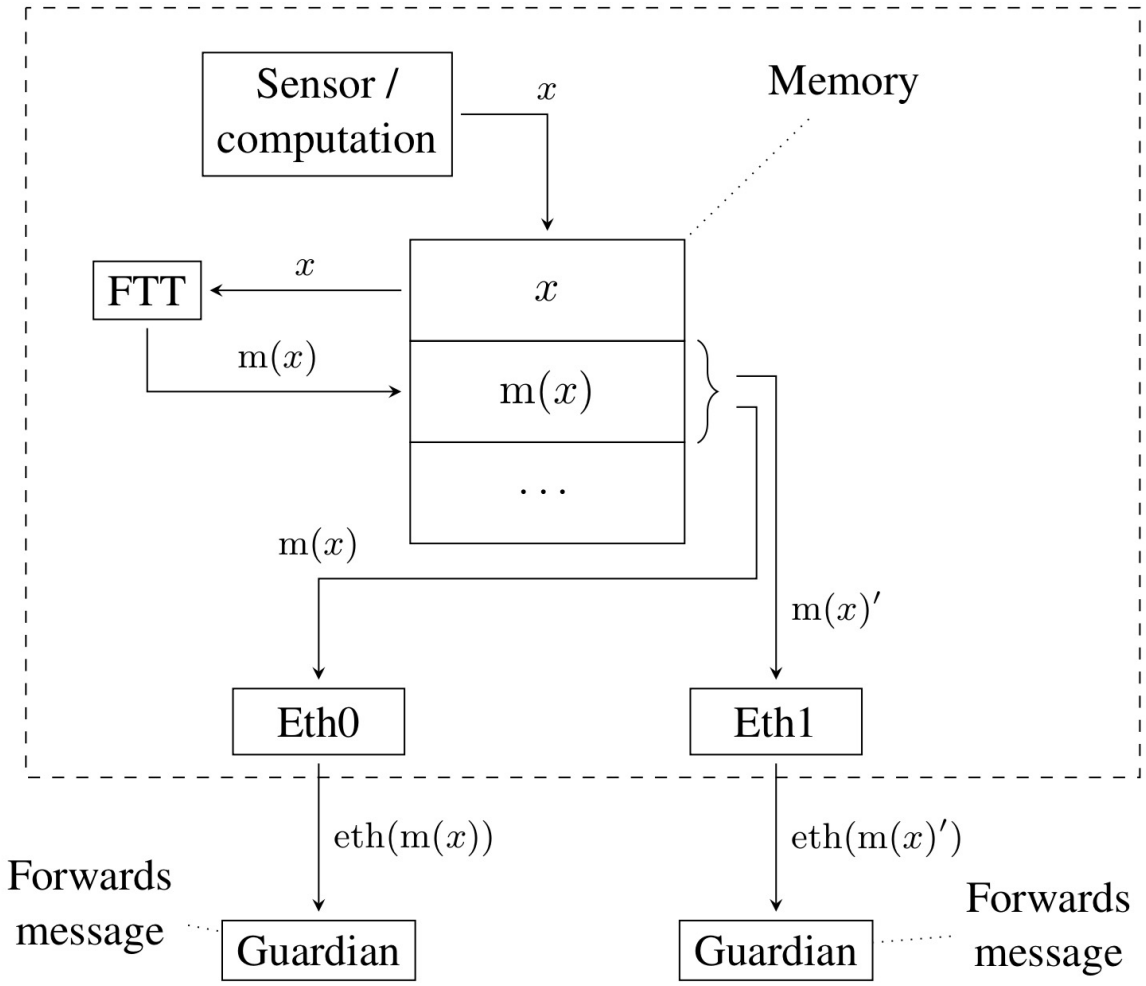
Channel (Spatial) Replication

- **Restriction of <u>slave</u> f-semantics: elimination of 2-faced**
  - Each node has 2 links and could send different replicas of a msg



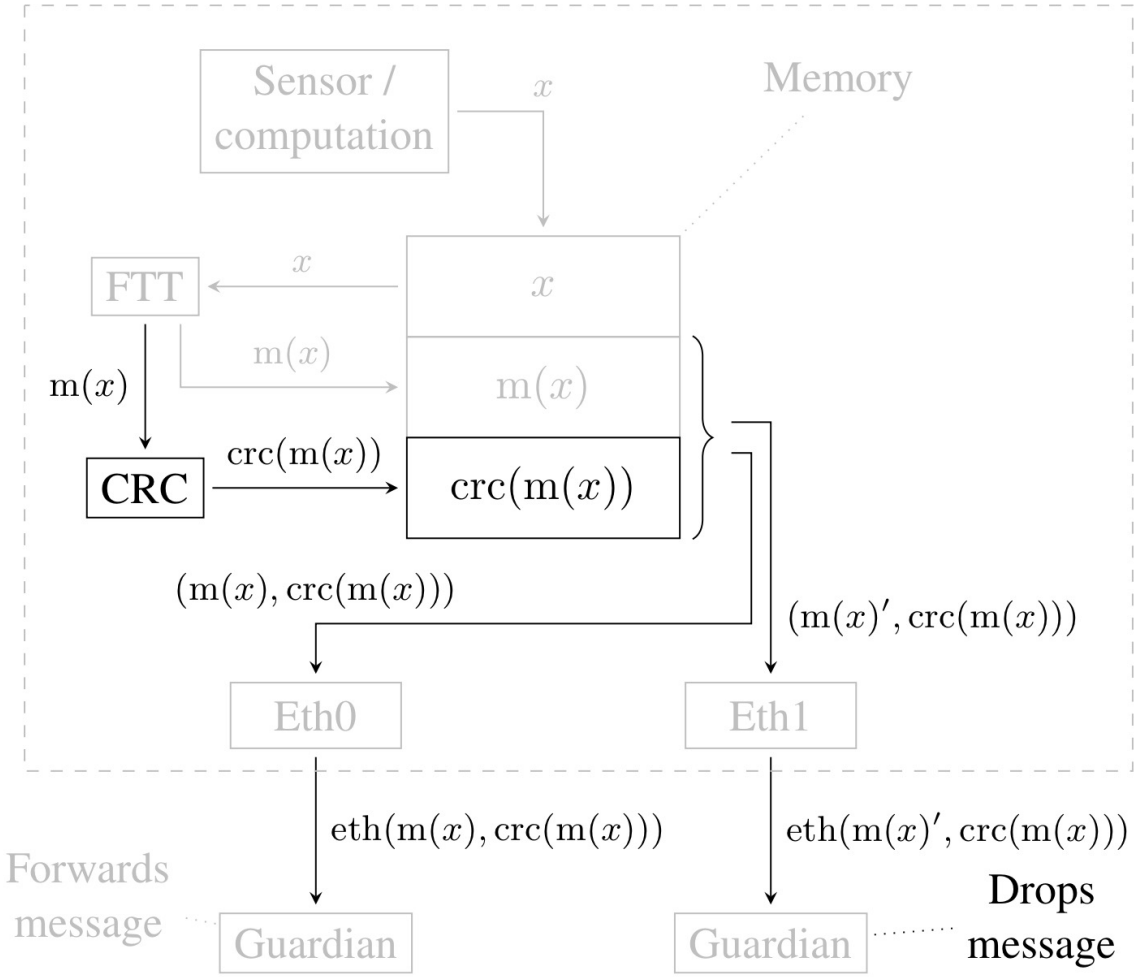
Channel (Spatial) Replication

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



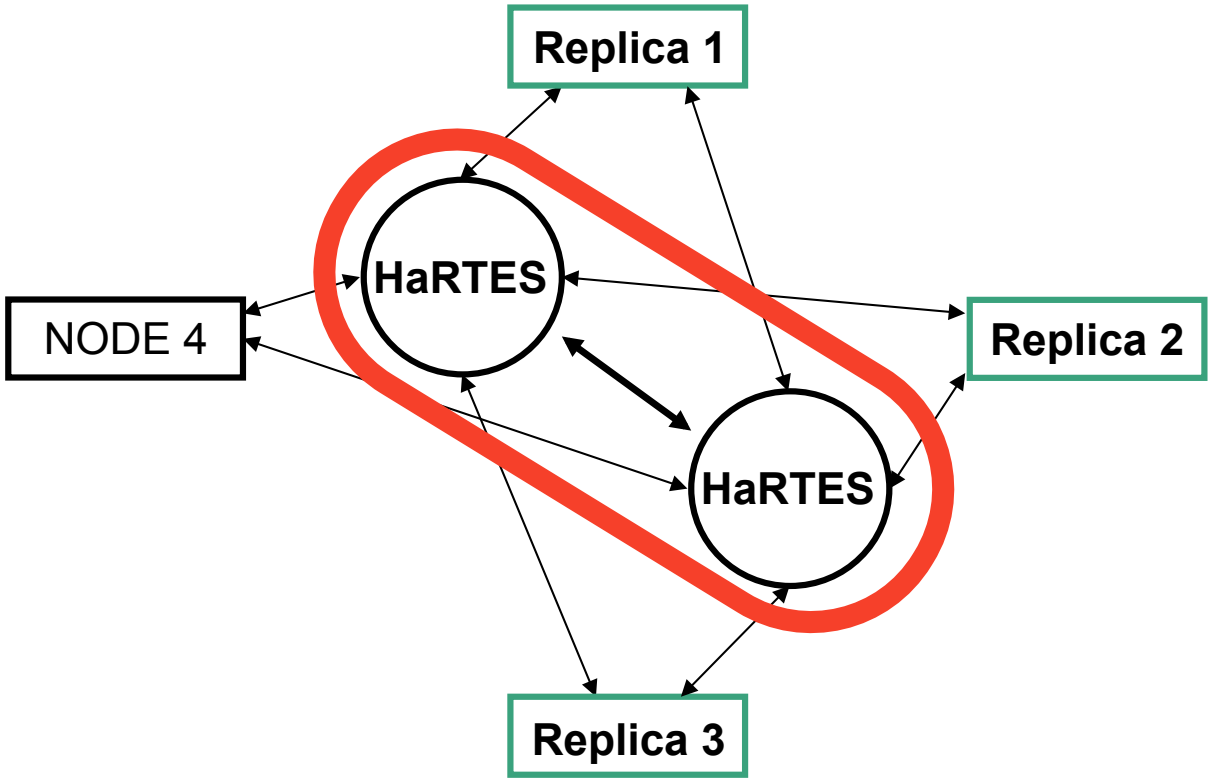
Channel (Spatial) Replication

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



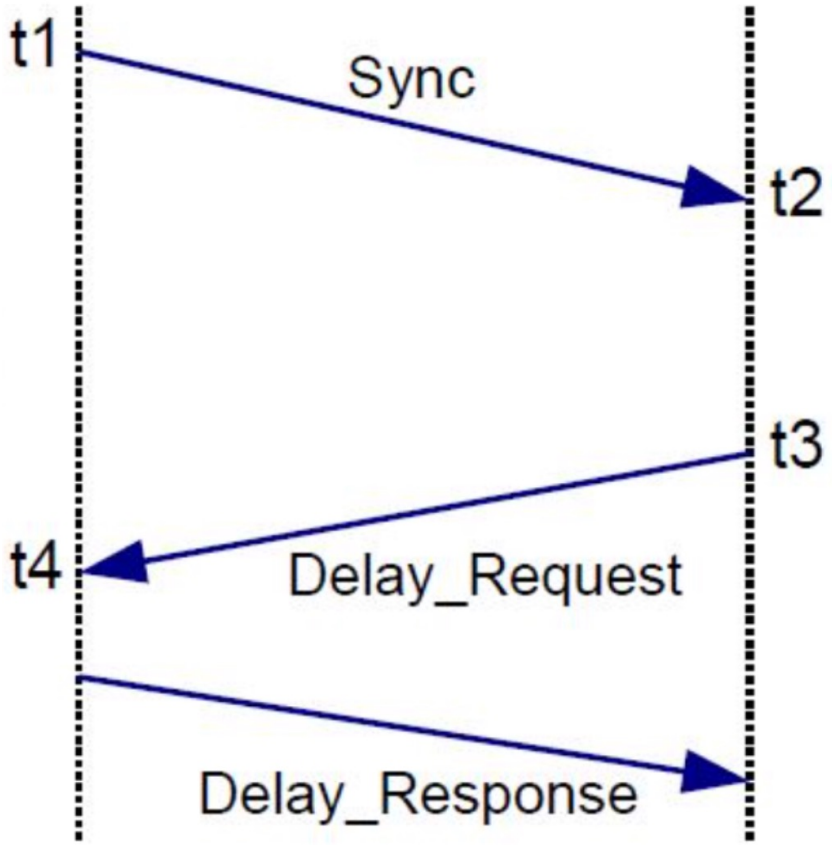
Channel (Spatial) Replication

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



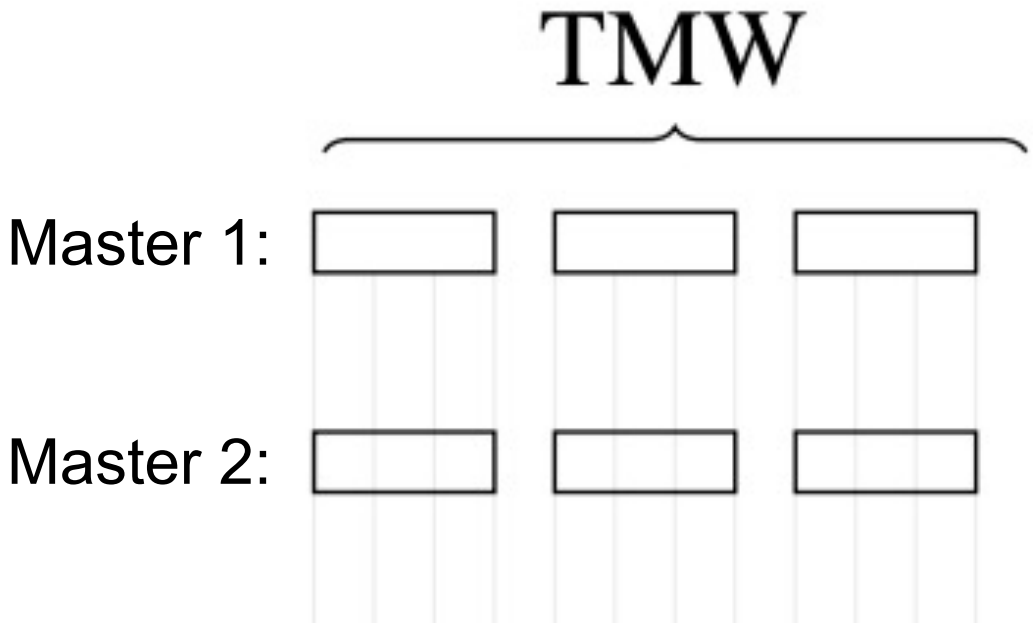
Channel (Spatial) Replication

- **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 (<u>value</u> 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

•Slave

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

•Main concerns

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

Demos

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