Designing **fault-diagnosis** and **reintegration** to **prevent node redundancy attrition** in highly reliable control systems based on **FTT-Ethernet**

Sinisa Derasevic, Manuel Barranco, Julián Proenza

Mathematics and Computer Science Department,
University of the Balearic Islands (UIB), Spain

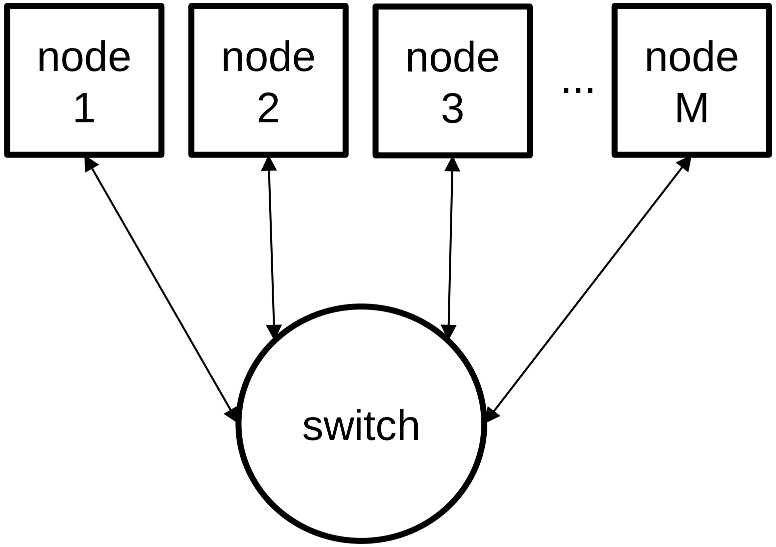
UIB
Universitat
de les Illes Balears

MINISTERIO
DE ECONOMÍA
Y COMPETITIVIDAD

Fondo Europeo de
Desarrollo Regional

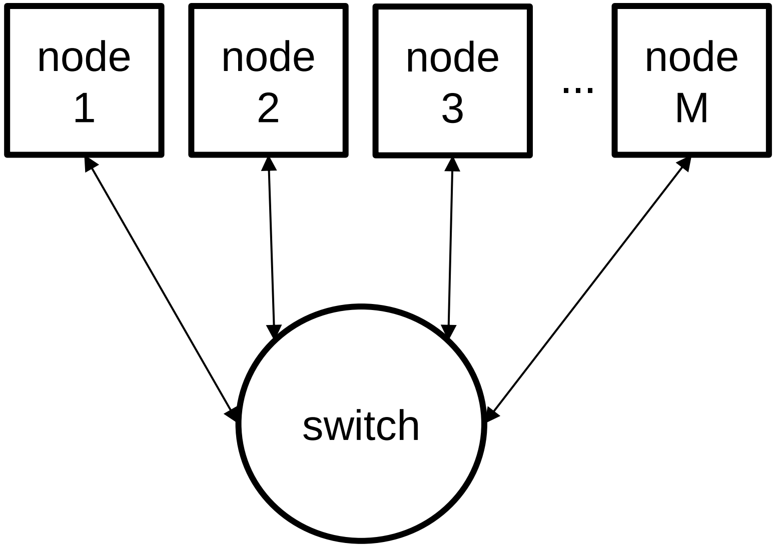
EUROWEB
European Research and Education Collaboration
with Western Balkan

diagnosis and **reintegration** of faulty **nodes** in **highly reliable** Distributed Control Systems based on **FTT-Ethernet**

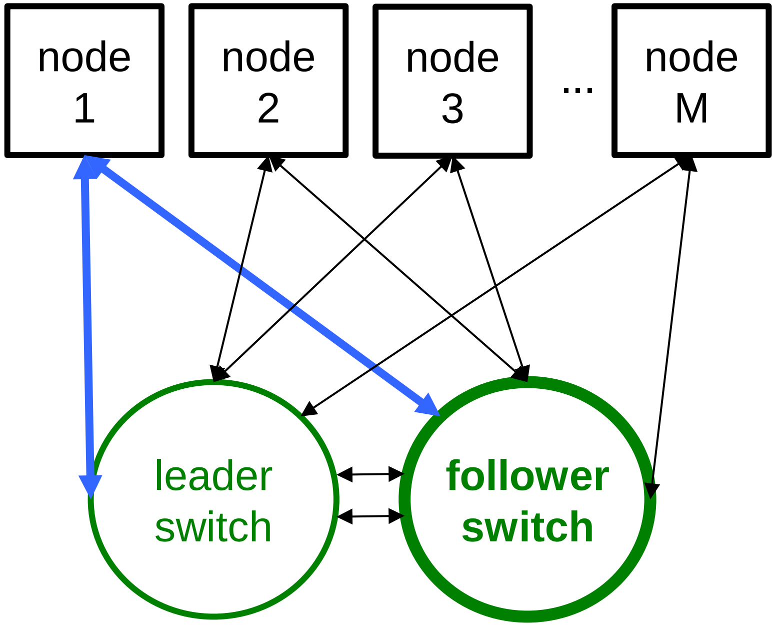


diagnosis and **reintegration** of faulty **nodes** in **highly reliable** Distributed Control Systems based on **FTT-Ethernet**

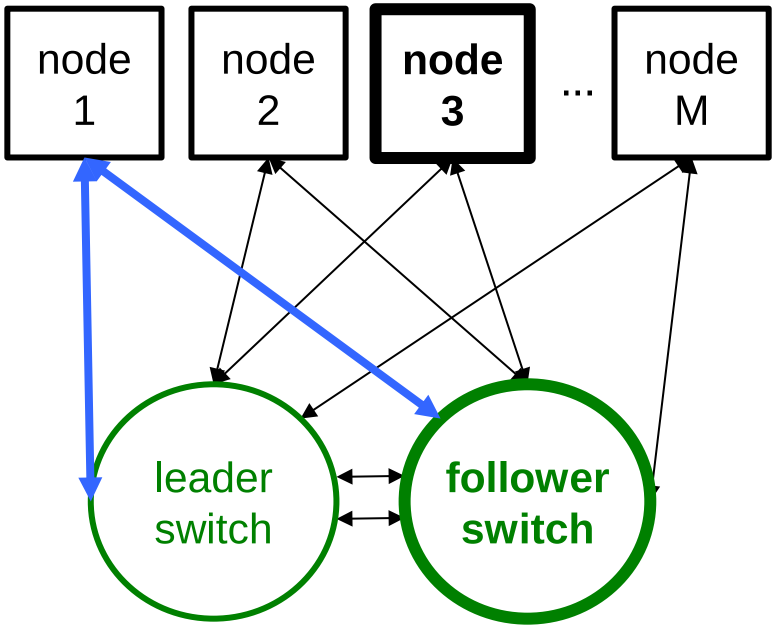
relevant piece of FT4FTT



- high reliability by tolerating faults at
  - switch → duplicate
  - links → duplicate
  - nodes

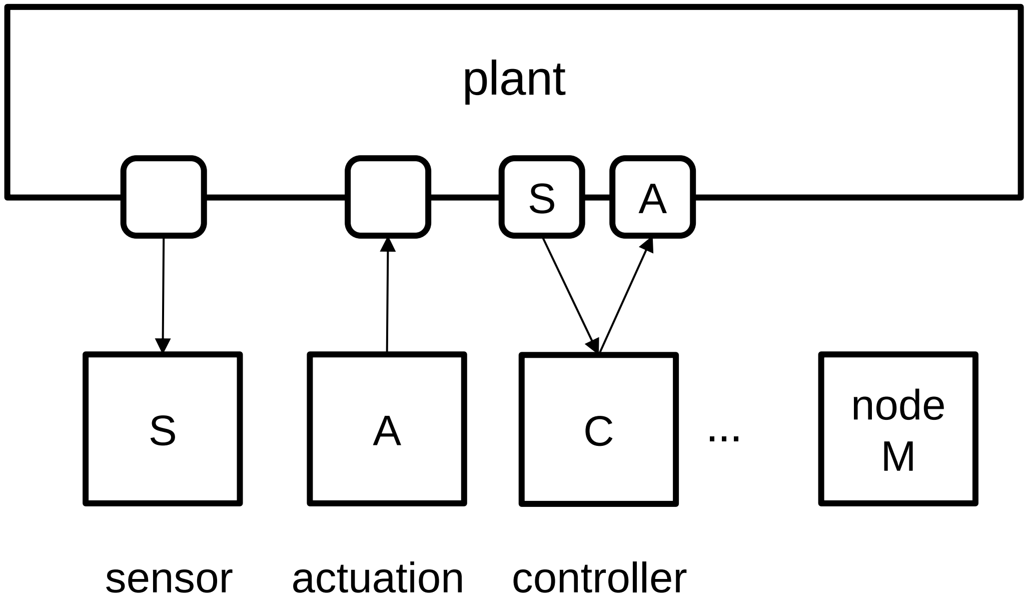


- high reliability by tolerating faults at
  - switch → duplicate
  - links → duplicate
  - nodes → **actively replicate critical** nodes & **vote**



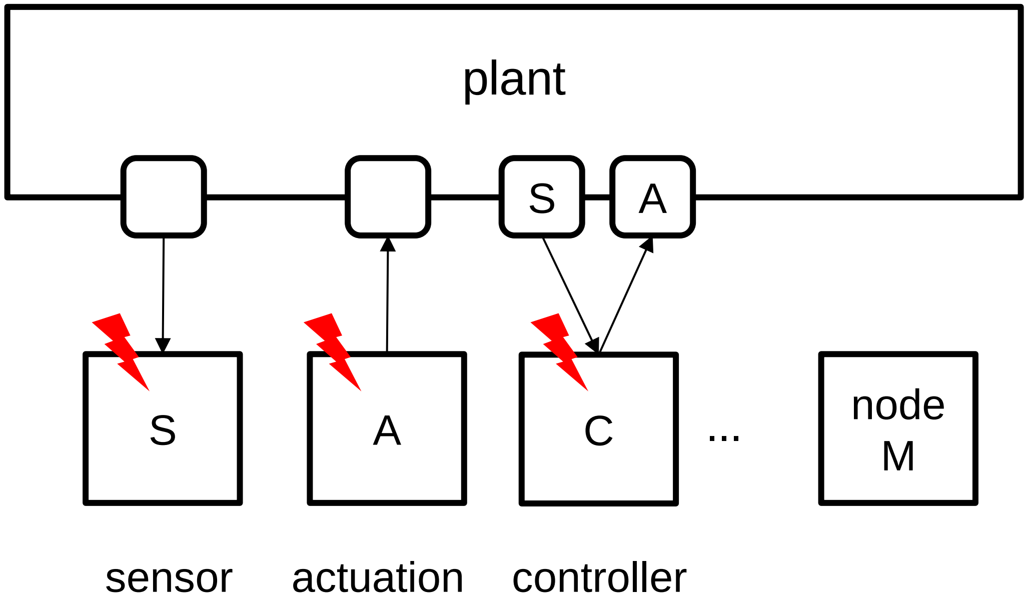
which are the critical nodes?

which are the critical nodes?



which are the critical nodes?

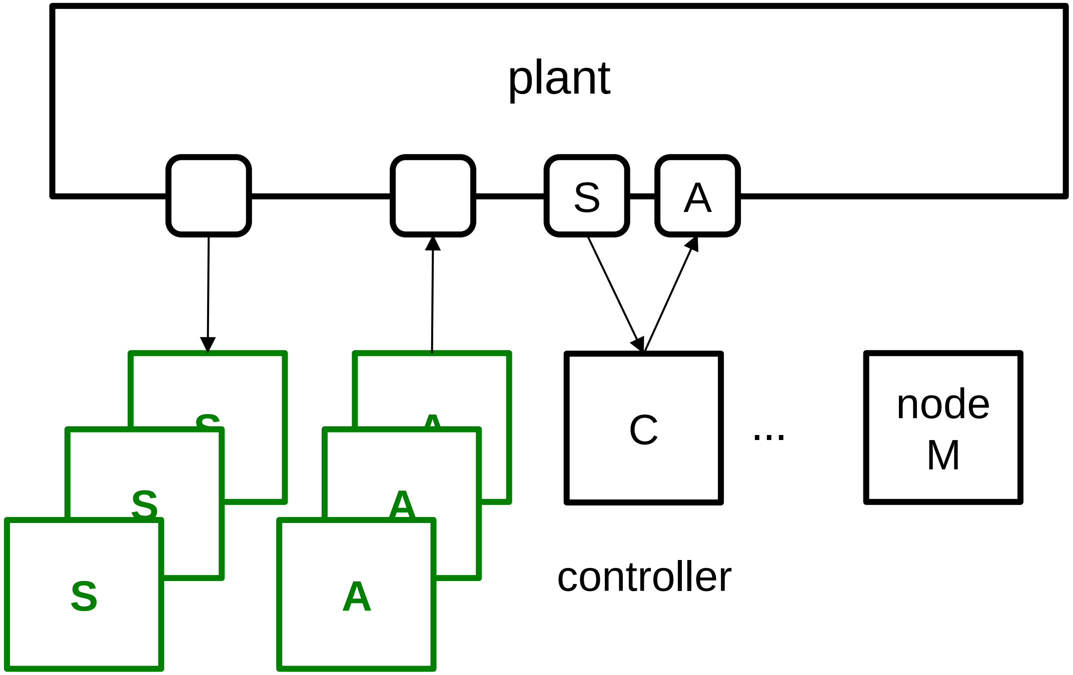
in principle all these nodes can be considered as critical



system failure

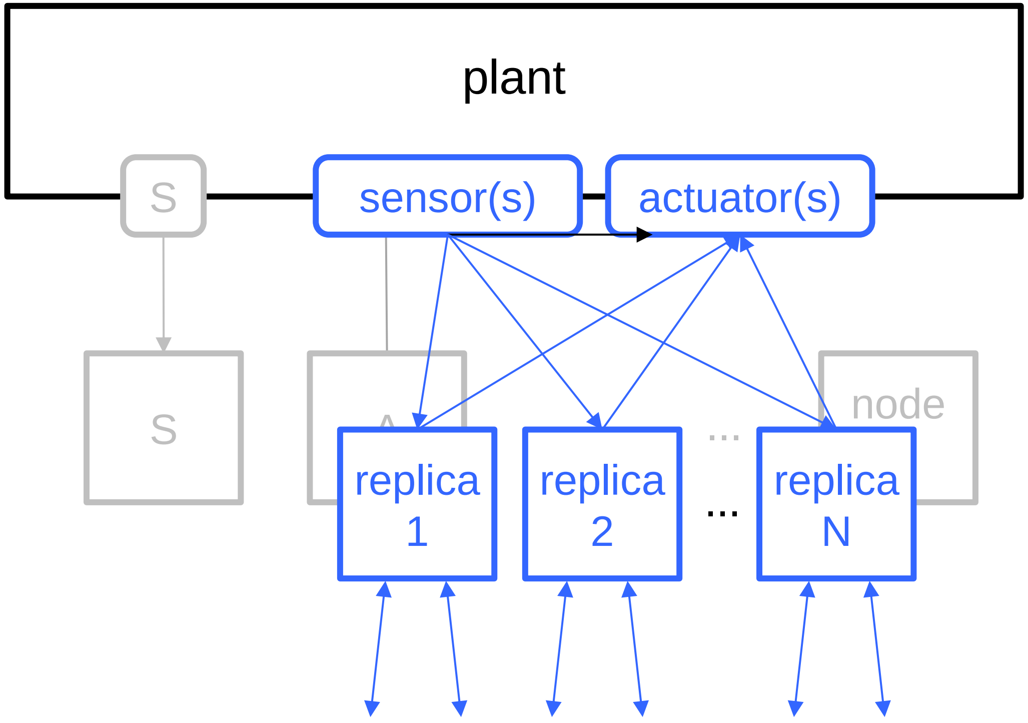
which are the critical nodes?

replicate **sensor** and **actuation** nodes is **trivial**



which are the critical nodes?

replicate a **controller** node is **complex:** replicas must **coordinate** among them



coordinate among them

how do replicas coordinate?

- **synchronize** at **communication** & **app.** levels
    - using the Trigger Message (TM)
- **vote** on **intermediate** results

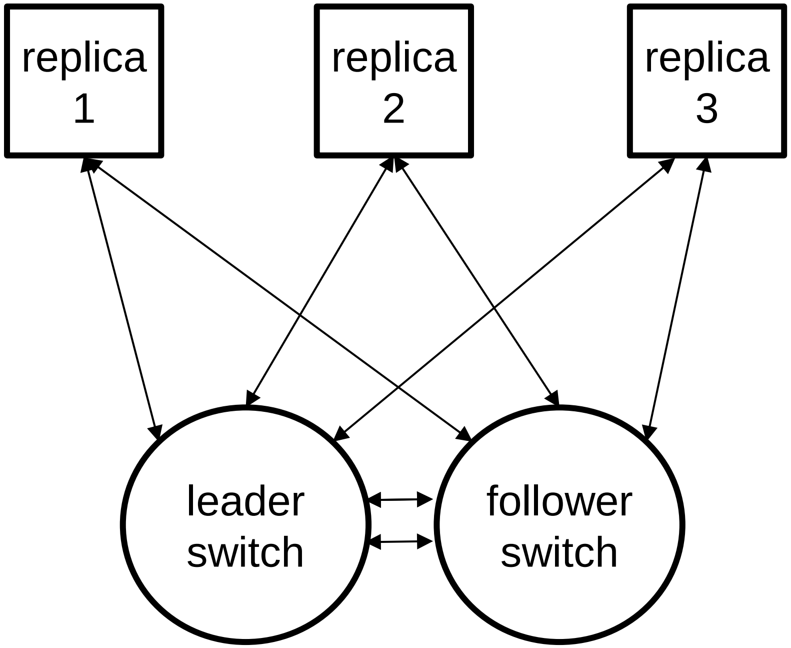
how do replicas coordinate?

- **synchronize** at **communication** & **app.** levels
  - using the Trigger Message (TM)
- **vote** on **intermediate** results ←

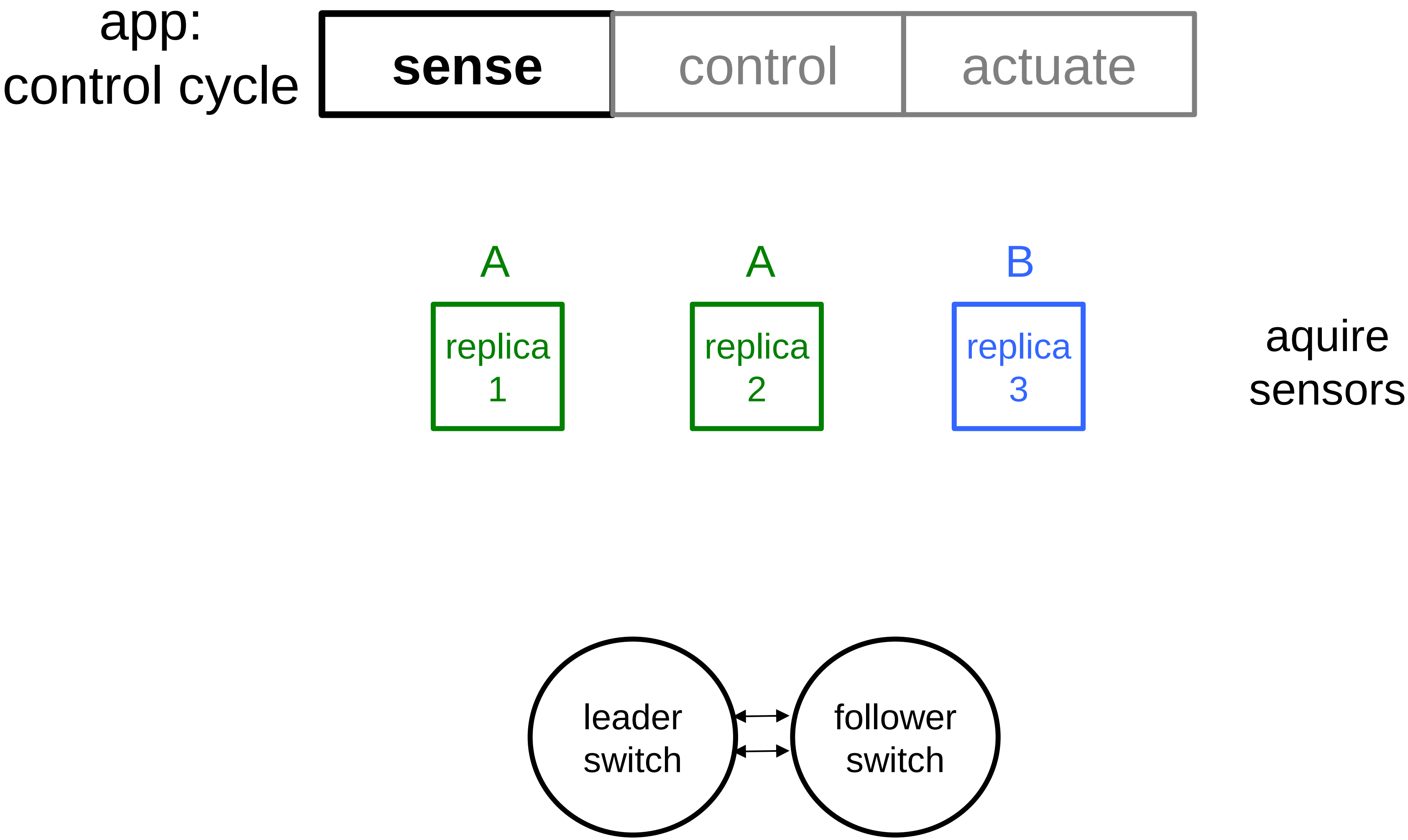
voting

app:
control cycle

sense	control	actuate



voting



app:
control cycle

sense control actuate

A A B

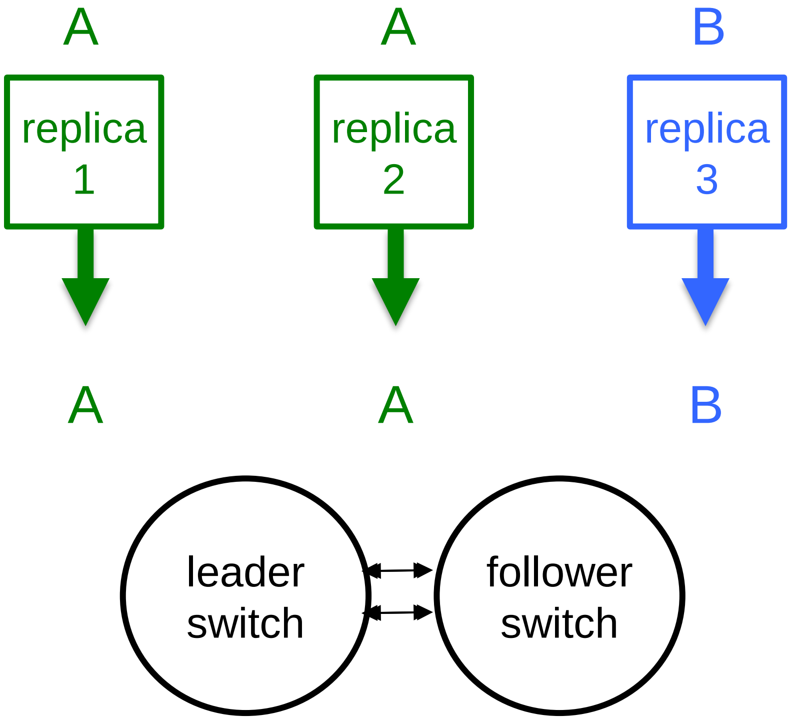
replica 1 replica 2 replica 3

aquire
sensors

leader switch

follower switch

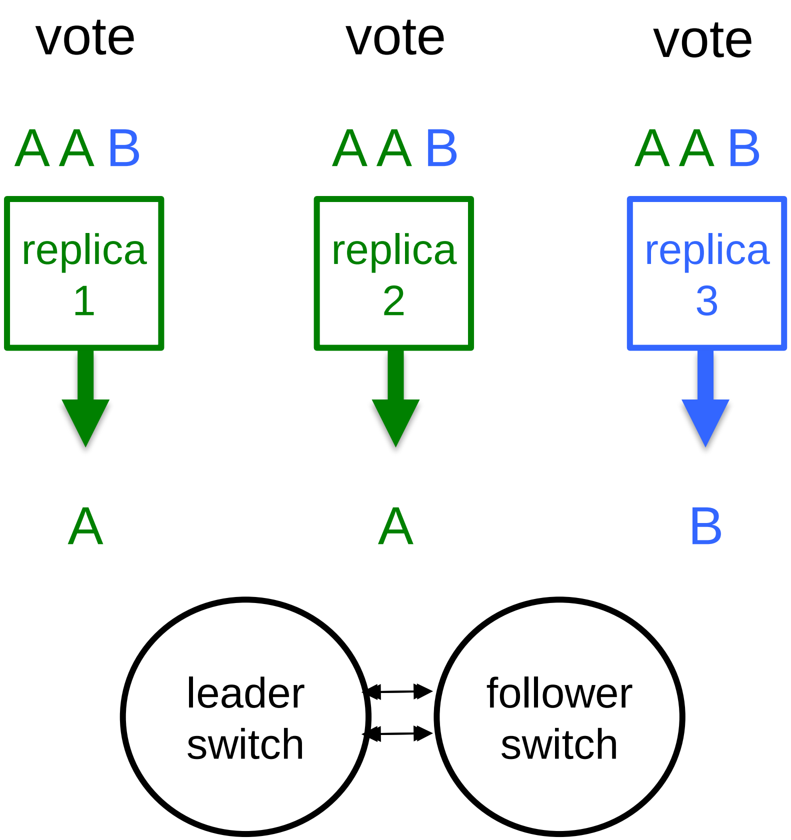
voting



aquire sensors

exchange sensors

voting

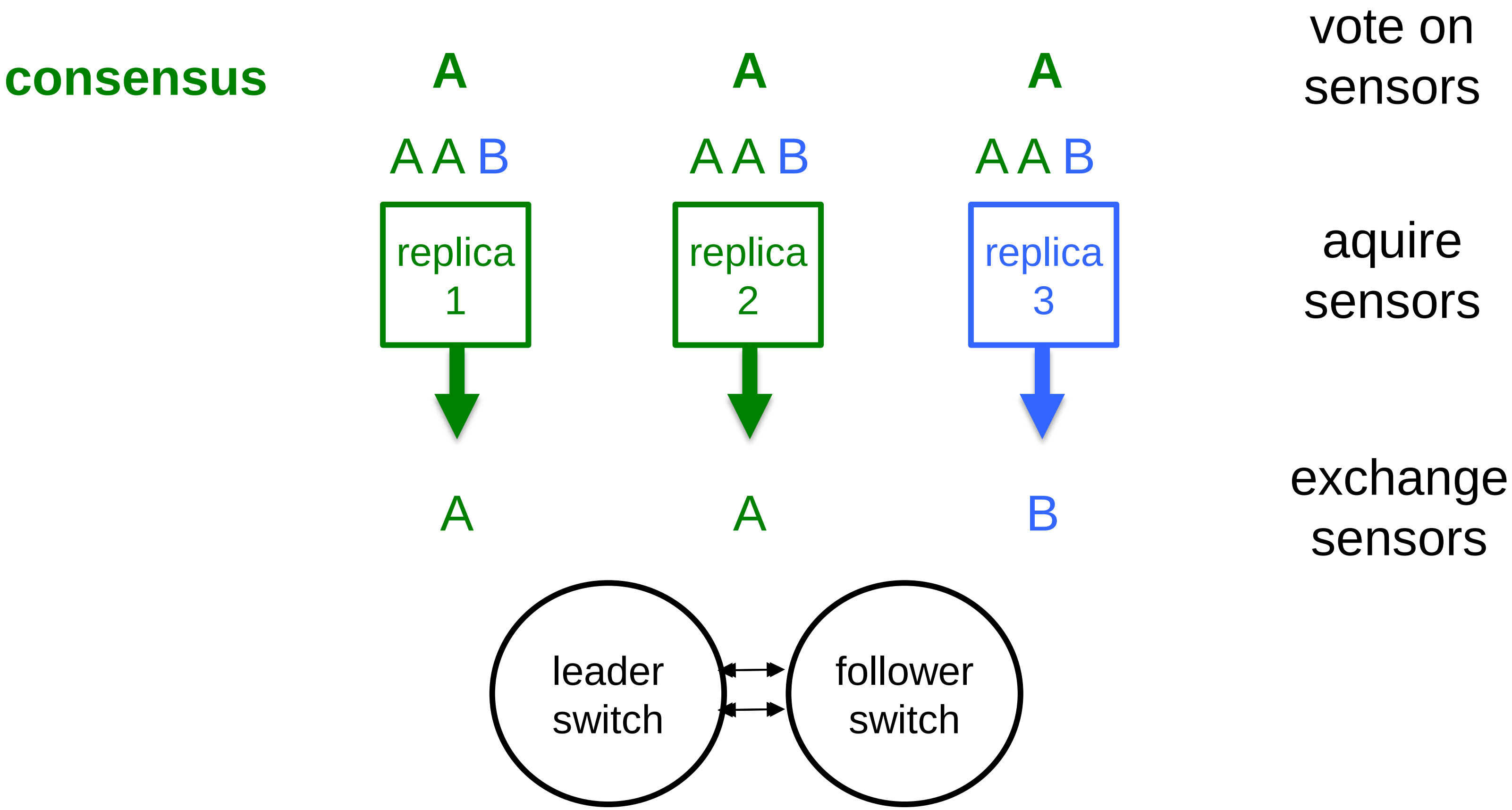


vote on sensors

aquire sensors

exchange sensors

voting



consensus

A A A

vote on sensors

A A B A A B A A B

replica 1

replica 2

replica 3

aquire sensors

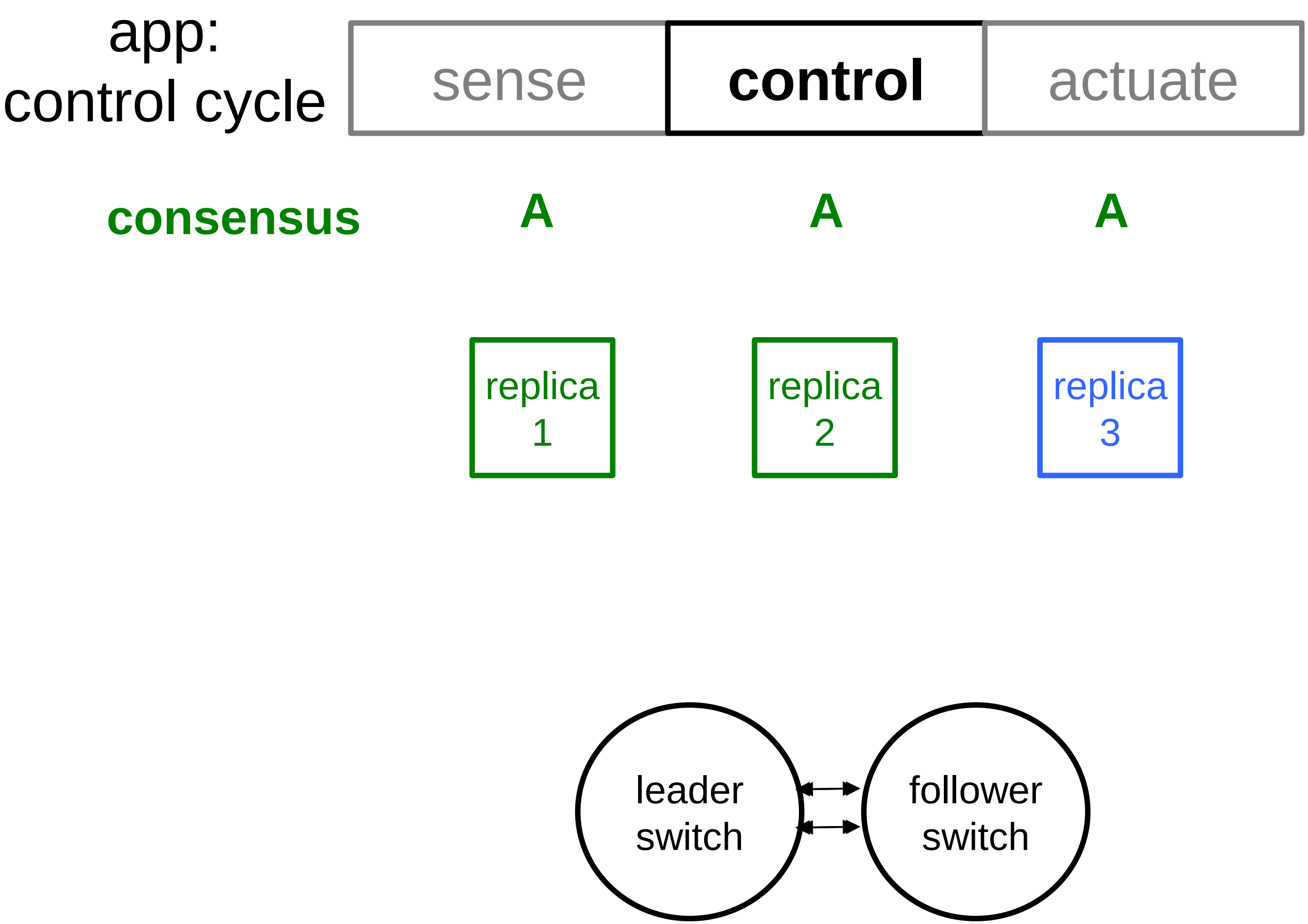
A A B

exchange sensors

leader switch

follower switch

voting



app:
control cycle

sense

control

actuate

consensus

A **A** **A**

replica 1

replica 2

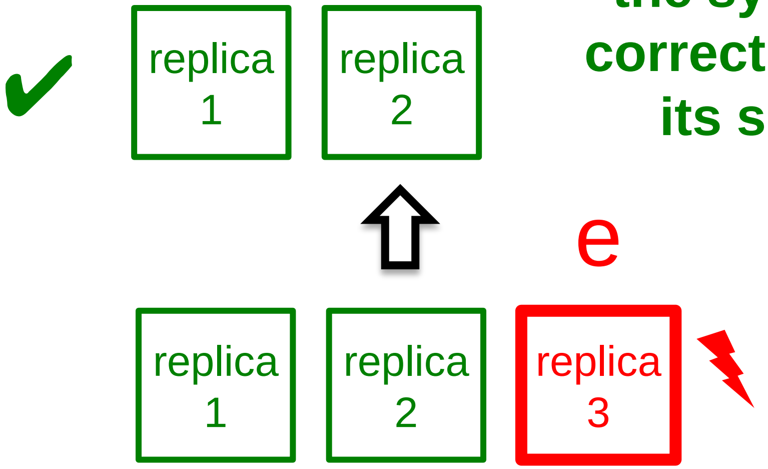
replica 3

leader switch

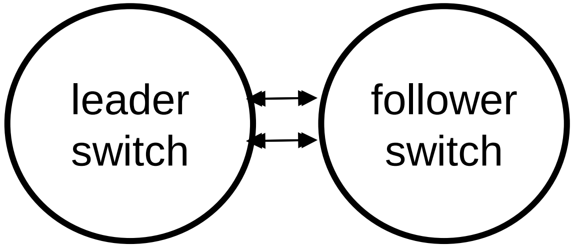
follower switch

benefits of active node replication with voting ?

compensate errors

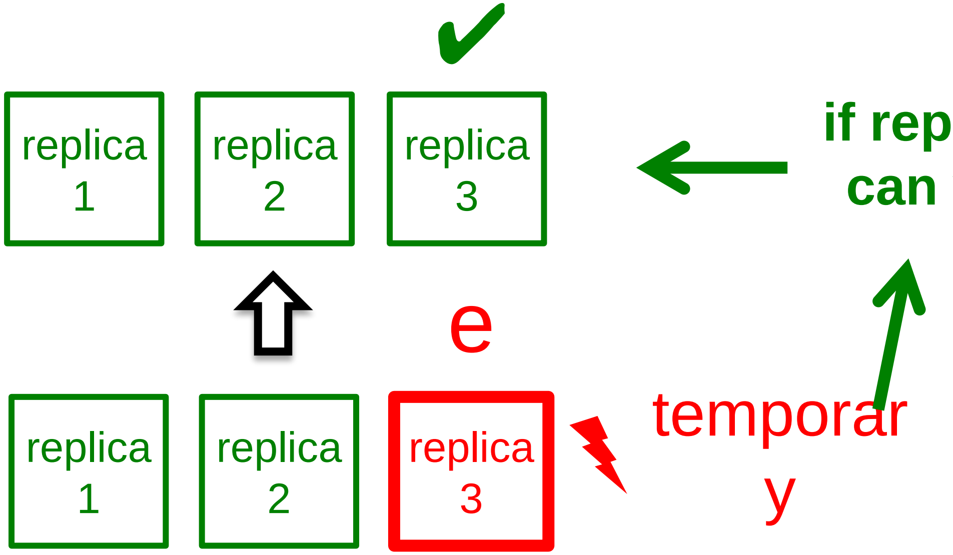


the sytem can correctly deliver its service

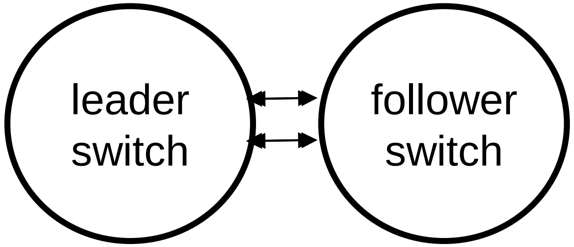


replicas may recover from errors

replica 3 recovers and keeps participating

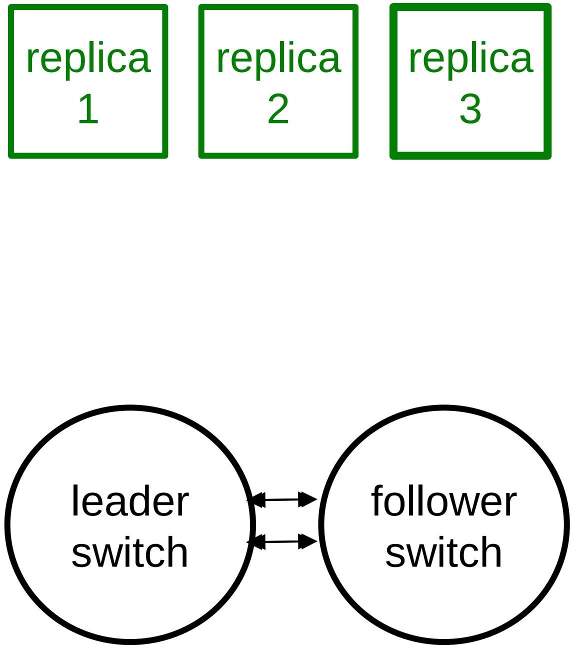


if replica 3 can vote



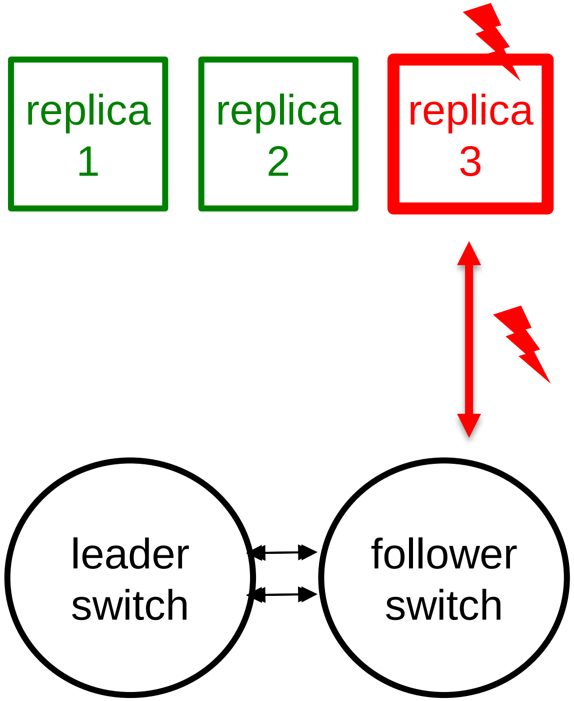
however…

what if a **temporary fault** makes a **replica** to be **lost from then on** ??



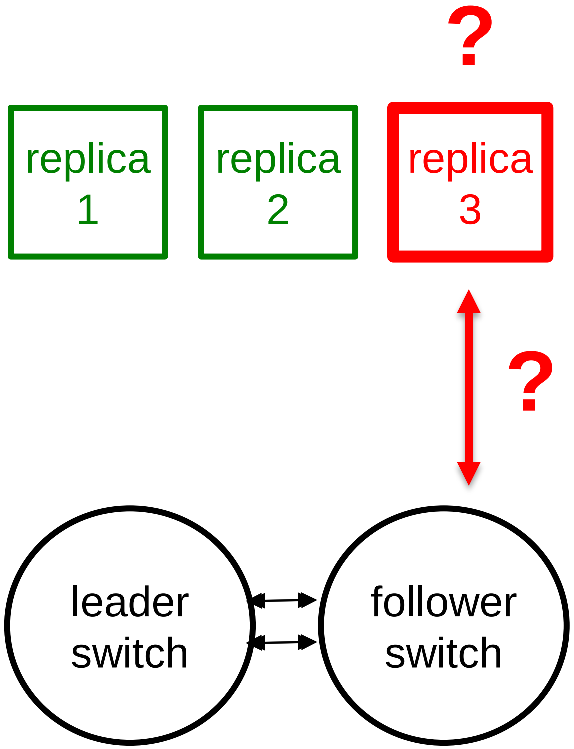
what if a **temporary fault** makes a **replica** to be **lost from then on** ??

temporary fault affects **replica** 3
internals or **communication capabilities**



what if a **temporary fault** makes a **replica** to be **lost from then on** ??

temporary fault affects **replica** 3
internals or **communication capabilities**

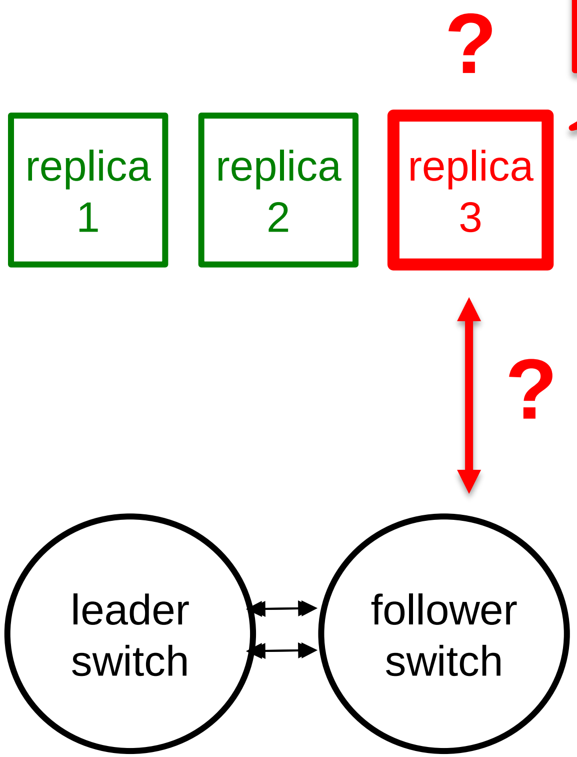


replica 3 may
desynchronize at the
level
of **application** and/or
communication

what if a **temporary fault**
makes a **replica** to be **lost from then on** ??

temporary fault affects **replica** 3
internals or **communication capabilities**

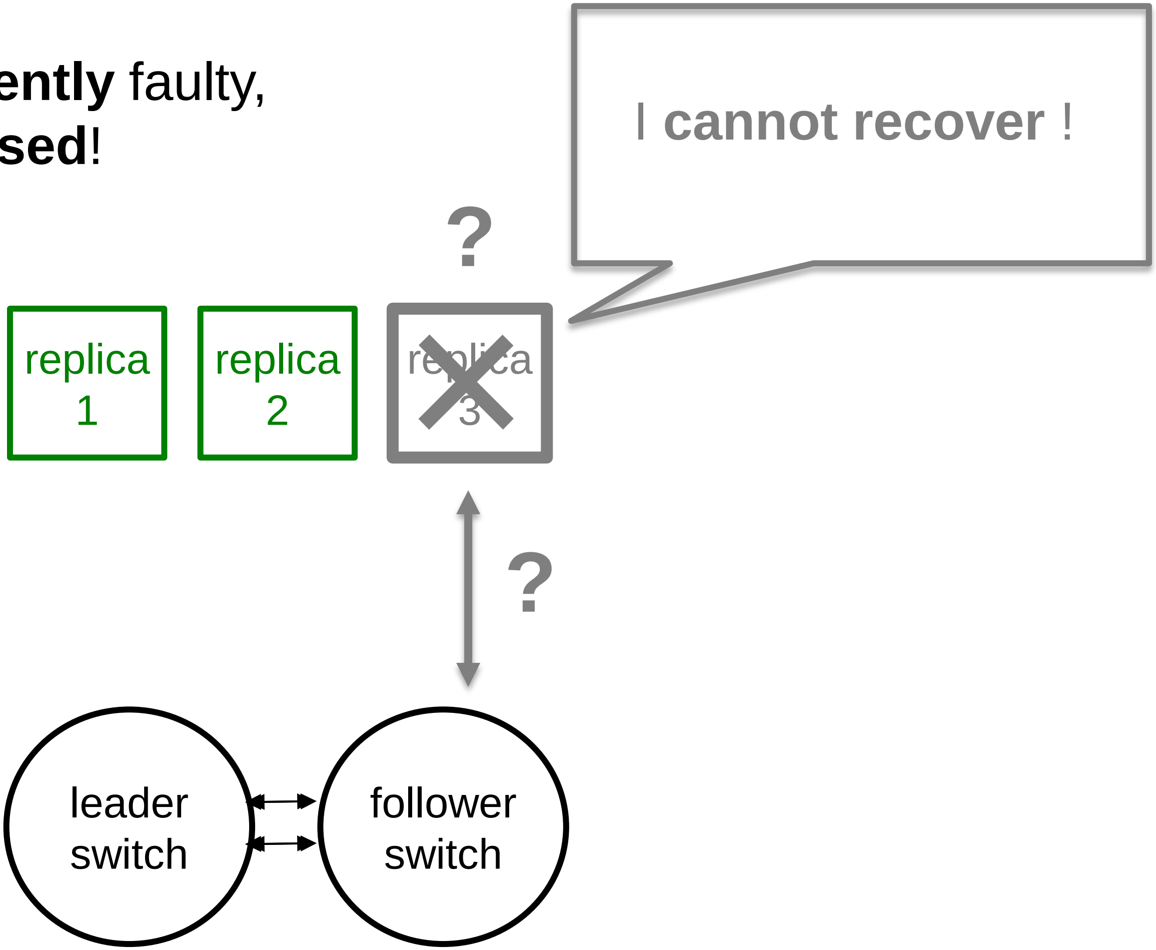
I cannot recover !



replica 3 may
desynchronize at the
level
of **application** and/or
communication

node redundancy attrition

replica 3 is **not permanently** faulty,
but can **not** be **used**!



temporary faults are more probable than permanent ones if we do not prevent redundancy attrition caused by temporary faults

then we do not
take full advantage
of the redundancy
investment

objective

prevent
node redundancy attrition

objective

identify and **implement** mechanisms to **diagnose** and **reintegrate** temporary-faulty nodes that are lost

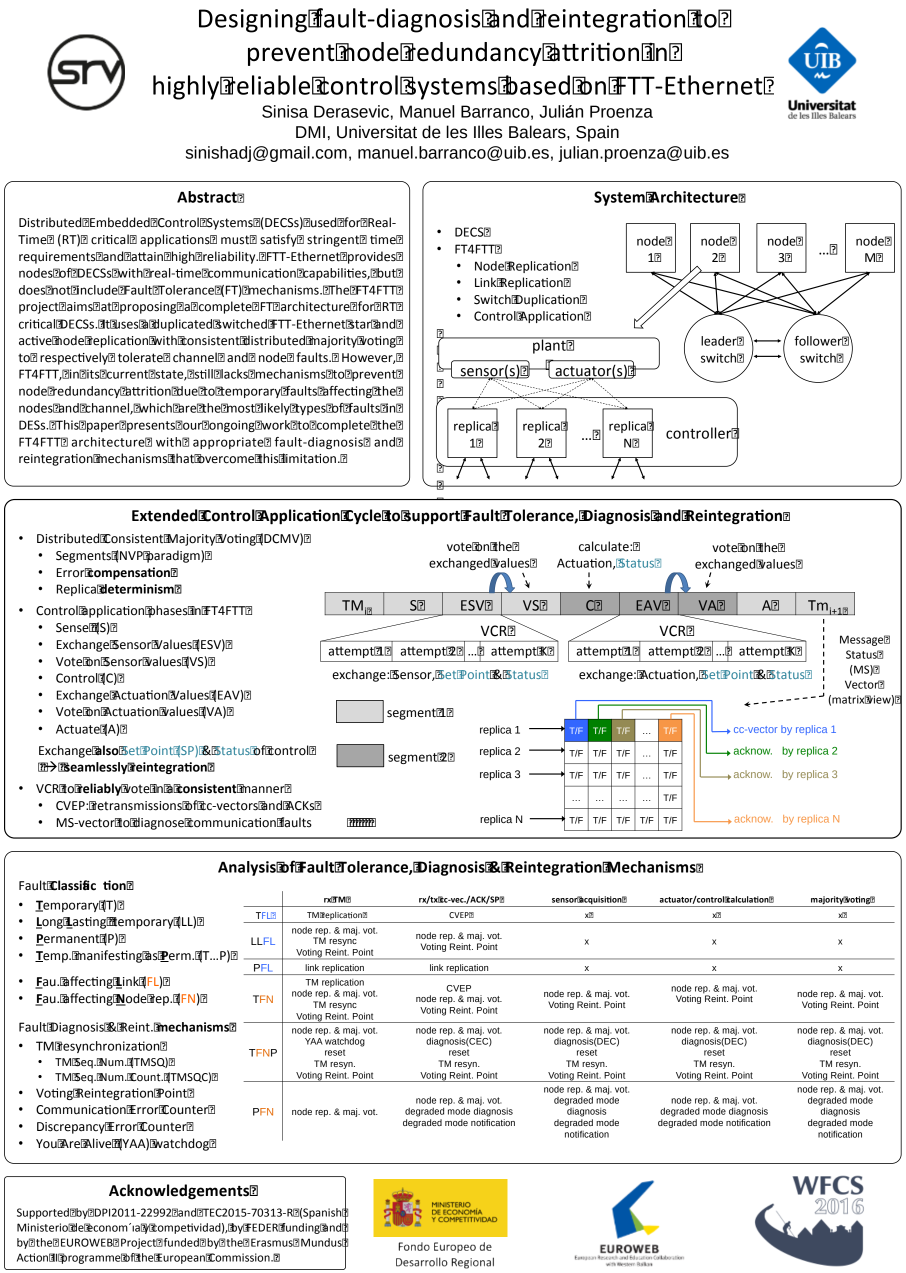
steps

- classify faults
- exhaustively analyze how they can affect a replica
- design needed mechanisms
- implement and test them

steps

- classify faults
- exhaustively analyze how they can affect a replica
- design needed mechanisms
- **implement and test them ← pending**

we plan to quantify the reliability improvement



Designing fault-diagnosis and reintegration to prevent node redundancy attrition in highly reliable control systems based on FTT-Ethernet

Sinisa Derasevic, Manuel Barranco, Julián Proenza
DMI, Universitat de les Illes Balears, Spain
sinishadj@gmail.com, manuel.barranco@uib.es, julian.proenza@uib.es

Abstract

Distributed Embedded Control Systems (DECSs) used for Real-Time (RT) critical applications must satisfy stringent time requirements and attain high reliability. FTT-Ethernet provides nodes of DECSs with real-time communication capabilities, but does not include Fault Tolerance (FT) mechanisms. The FT4FTT project aims at proposing a complete FT architecture for RT critical DECSs. It uses a duplicated switched FTT-Ethernet star and active node replication with consistent distributed majority voting to respectively tolerate channel and node faults. However, FT4FTT, in its current state, still lacks mechanisms to prevent node redundancy attrition due to temporary faults affecting the nodes and channel, which are the most likely types of faults in DESs. This paper presents our ongoing work to complete the FT4FTT architecture with appropriate fault-diagnosis and reintegration mechanisms that overcome this limitation.

System Architecture

- DECS
- FT4FTT
  - Node Replication
  - Link Replication
  - Switch Duplication
  - Control Application

node 1
node 2
node 3
...
node M
leader switch
follower switch
plant
sensor(s)
actuator(s)
replica 1
replica 2
...
replica N
controller

Extended Control Application Cycle to support Fault Tolerance, Diagnosis and Reintegration

- Distributed Consistent Majority Voting (DCMV)
  - Segments (NVP paradigm)
  - Error **compensation**
  - Replica **determinism**
- Control application phases in FT4FTT
  - Sense (S)
  - Exchange Sensor Values (ESV)
  - Vote on Sensor values (VS)
  - Control (C)
  - Exchange Actuation Values (EAV)
  - Vote on Actuation values (VA)
  - Actuate (A)

  Exchange **also** Set Point (SP) & Status of control
  → **seamlessly reintegration**
- VCR to **reliably** vote in a **consistent** manner
  - CVEP: retransmissions of cc-vectors and ACKs
  - MS-vector to diagnose communication faults

vote on the exchanged values
calculate: Actuation, Status
vote on the exchanged values

TM$_i$	S	ESV	VS	C	EAV	VA	A	Tm$_{i+1}$

VCR

attempt 1	attempt 2	...	attempt K

exchange: Sensor, Set Point & Status

VCR

attempt 1	attempt 2	...	attempt K

exchange: Actuation, Set Point & Status

Message Status (MS) Vector (matrix view)

▨ segment 1
▨ segment 2

replica 1	T/F	T/F	T/F	...	T/F
replica 2	T/F	T/F	T/F	...	T/F
replica 3	T/F	T/F	T/F	...	T/F
	...	...	...	...	T/F
replica N	T/F	T/F	T/F	T/F	T/F

cc-vector by replica 1
acknow. by replica 2
acknow. by replica 3
acknow. by replica N

Analysis of Fault Tolerance, Diagnosis & Reintegration Mechanisms

Fault **Classific tion**

- **T**emporary (T)
- **L**ong **L**asting temporary (LL)
- **P**ermanent (P)
- **T**emp. manifesting as **P**erm. (T…P)
- **F**au. affecting **L**ink (FL)
- **F**au. affecting **N**ode rep. (FN)

Fault Diagnosis & Reint. **mechanisms**

- TM resynchronization
  - TM Seq. Num. (TMSQ)
  - TM Seq. Num. Count. (TMSQC)
- Voting Reintegration Point
- Communication Error Counter
- Discrepancy Error Counter
- You Are Alive (YAA) watchdog

	rx TM	**rx/tx cc-vec./ACK/SP**	**sensor acquisition**	**actuator/control calculation**	**majority voting**
TFL	TM replication	CVEP	x	x	x
LLFL	node rep. & maj. vot. TM resync Voting Reint. Point	node rep. & maj. vot. Voting Reint. Point	x	x	x
PFL	link replication	link replication	x	x	x
TFN	TM replication node rep. & maj. vot. TM resync Voting Reint. Point	CVEP node rep. & maj. vot. Voting Reint. Point	node rep. & maj. vot. Voting Reint. Point	node rep. & maj. vot. Voting Reint. Point	node rep. & maj. vot. Voting Reint. Point
TFNP	node rep. & maj. vot. YAA watchdog reset TM resyn. Voting Reint. Point	node rep. & maj. vot. diagnosis(CEC) reset TM resyn. Voting Reint. Point	node rep. & maj. vot. diagnosis(DEC) reset TM resyn. Voting Reint. Point	node rep. & maj. vot. diagnosis(DEC) reset TM resyn. Voting Reint. Point	node rep. & maj. vot. diagnosis(DEC) reset TM resyn. Voting Reint. Point
PFN	node rep. & maj. vot.	node rep. & maj. vot. degraded mode diagnosis degraded mode notification	node rep. & maj. vot. degraded mode diagnosis degraded mode notification	node rep. & maj. vot. degraded mode diagnosis degraded mode notification	node rep. & maj. vot. degraded mode diagnosis degraded mode notification

Acknowledgements

Supported by DPI2011-22992 and TEC2015-70313-R (Spanish Ministerio de econom´ıa y competividad), by FEDER funding and by the EUROWEB Project funded by the Erasmus Mundus Action II programme of the European Commission.

MINISTERIO DE ECONOMÍA Y COMPETITIVIDAD
Fondo Europeo de Desarrollo Regional

EUROWEB
European Research and Education Collaboration with Western Balkan

WFCS 2016

thank you for your attention !!