



---

# Improving Error Containment and Reliability of Controller Area Network (CAN) by means of Adequate Star Topologies

---

Manuel Barranco

Julián Proenza

Luis Almeida

# Introduction

## CAN (Controller Area Network) protocol

---

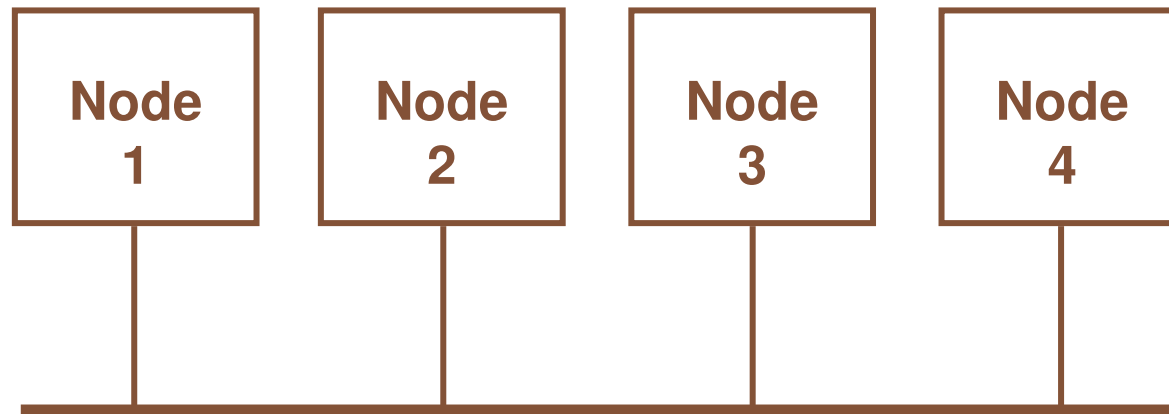
- **Field-bus** communication protocol mainly used in **distributed** control systems.
- **Extensively** used in practice for over 15 years in:
  - ✓ In-vehicle and intra-building communication.
  - ✓ Factory automation.
  - ✓ Some space applications.
- Main characteristics
  - ✓ Low cost.
  - ✓ Interesting real-time features.
  - ✓ **Good dependability.**

# Introduction

## CAN protocol - Basic properties

---

- Simplex **bus** topology.

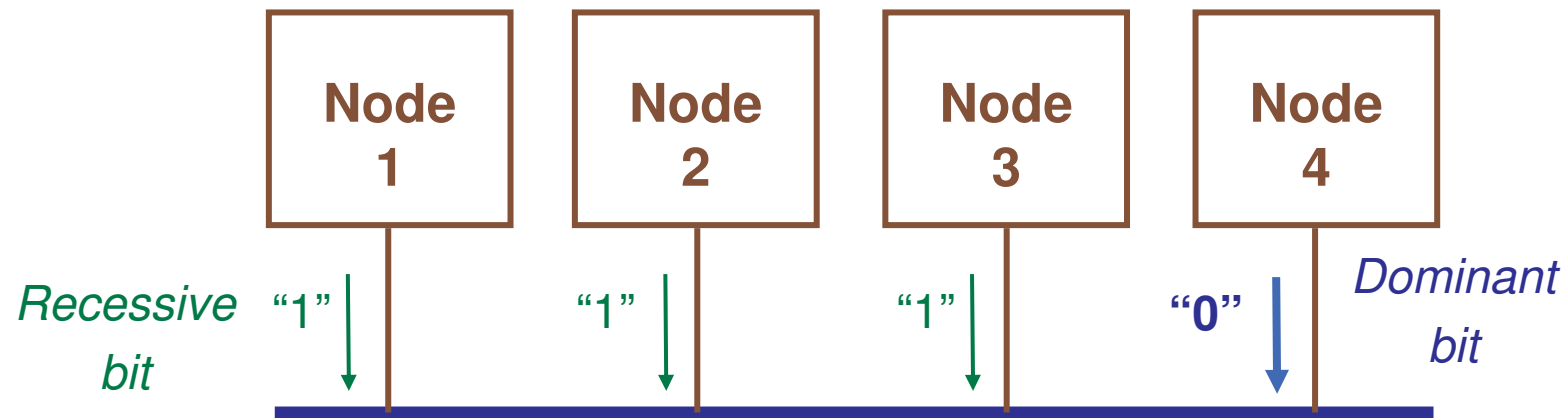


# Introduction

## CAN protocol - Basic properties

---

- **Dominant / recessive** transmission: the medium implements a **wired-AND** function.



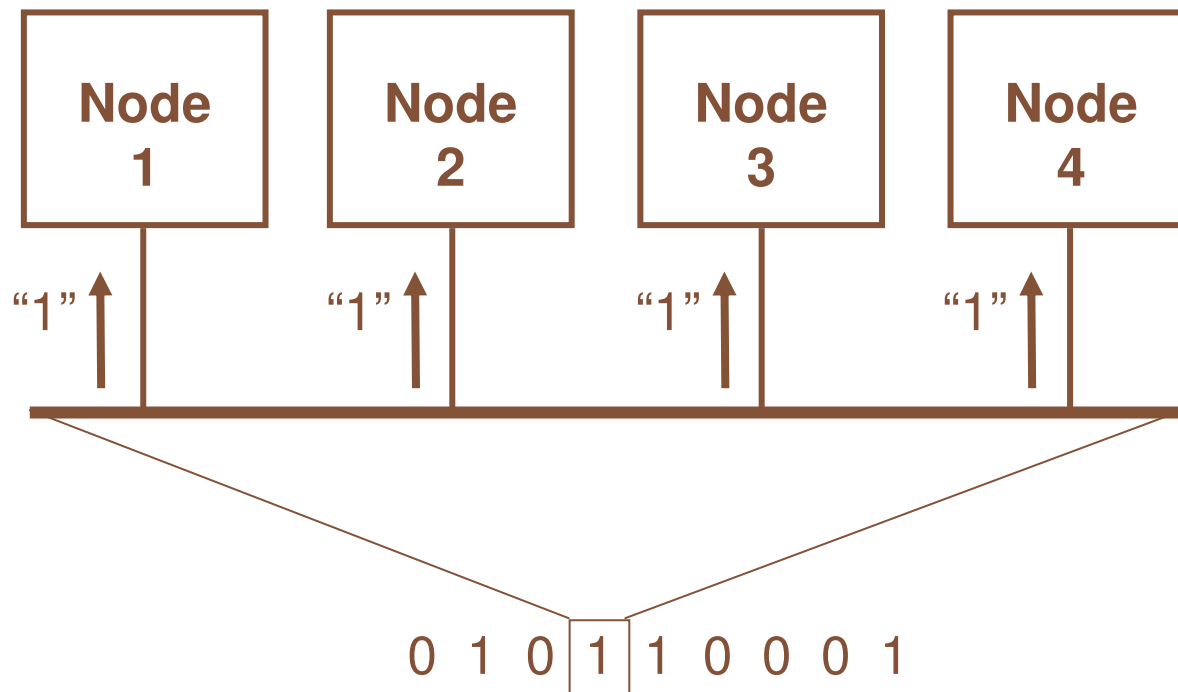
*Dominant bits overwrite recessive bits*

# Introduction

## CAN protocol - Basic properties

---

- **In-bit response:** nodes have a **quasi-simultaneous view** of every **bit** in the channel.

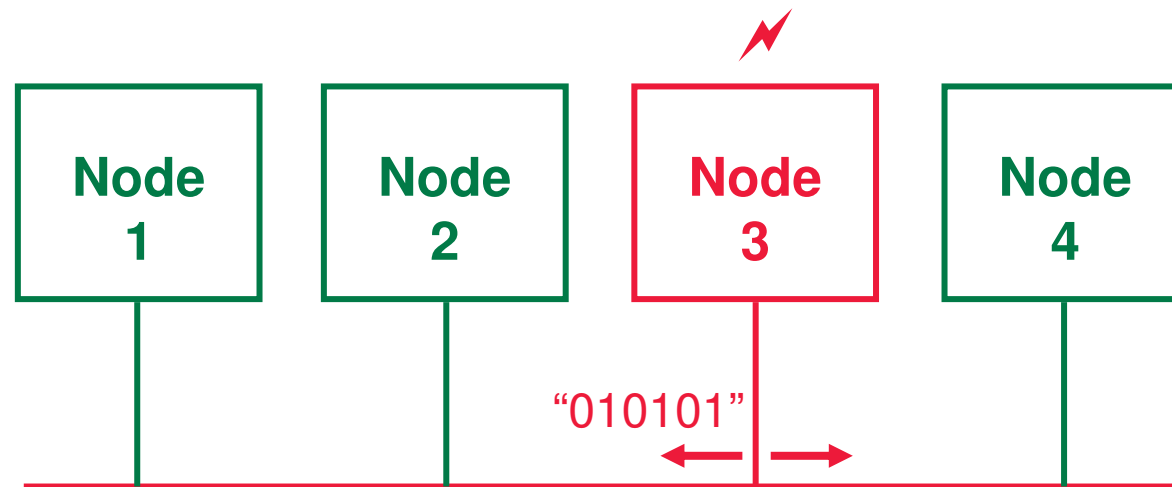


# Introduction

## CAN protocol - Basic properties

---

- **Fault-treatment** mechanisms.

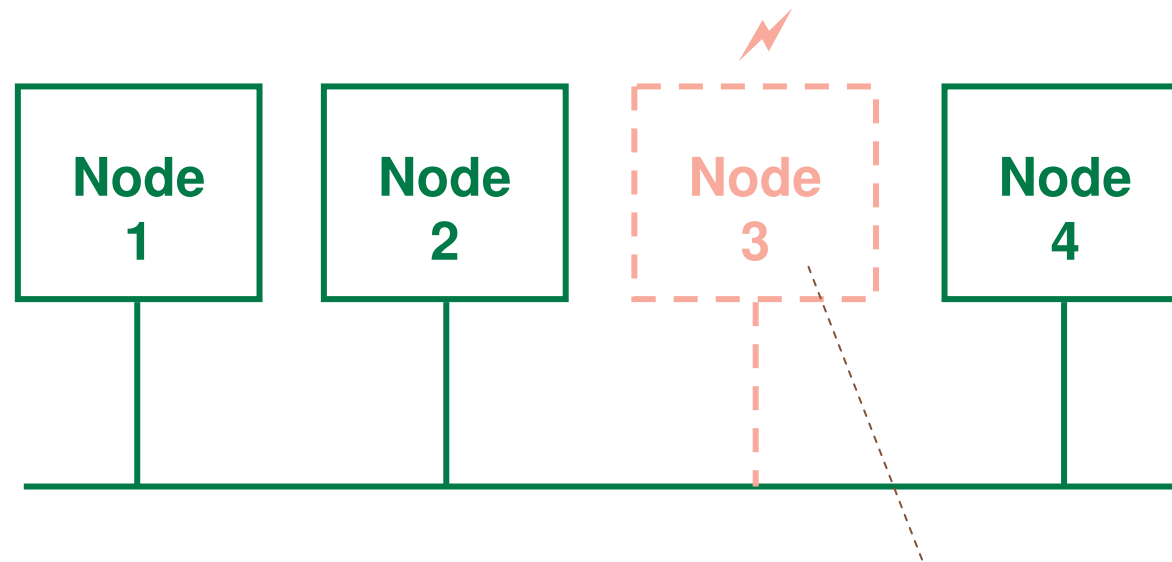


# Introduction

## CAN protocol - Basic properties

---

- **Fault-treatment** mechanisms.



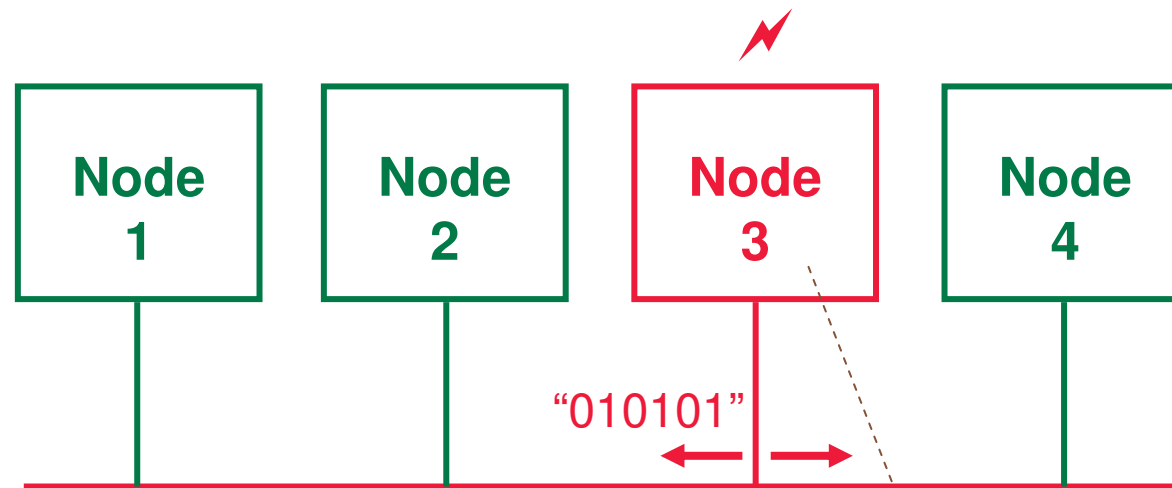
Node shuts down when it diagnoses itself  
as being permanently faulty

# Introduction

## CAN protocol – Scarce error containment

---

- A bus has **scarce error-containment** mechanisms.



If the node does not shut down when faulty, it **cannot prevent** the **propagation** of errors



# Introduction

## Formalization of the problem

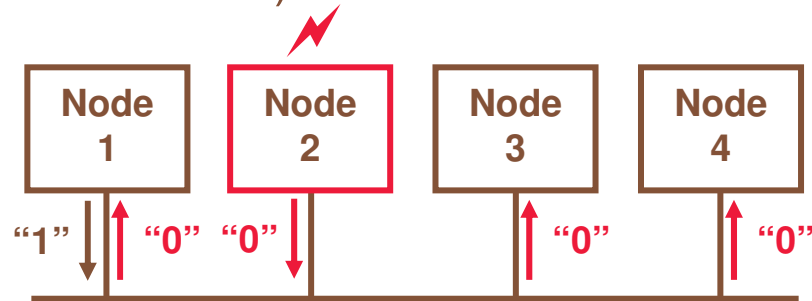
---

- K-severe failure of communication.
  - ✓ **Less than N-K nodes** of an ensemble of N nodes can **communicate with each other**.
- Point of k-severe failure of communication.
  - ✓ **Point whose failure** provokes a k-severe failure of communication.
  - ✓ It **includes** the concept of **single point of failure**.
  - ✓ A **bus** has **multiple** points of k-severe failure.

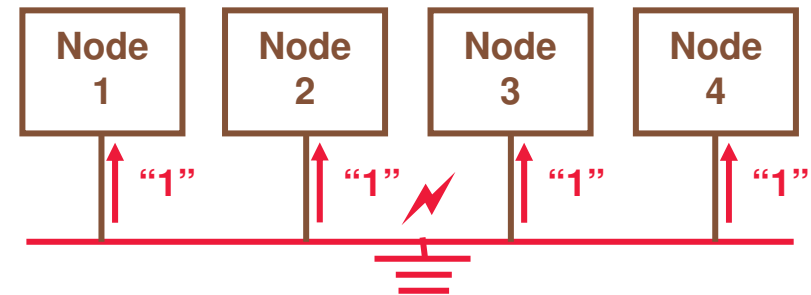
# Introduction

## Formalization of the problem – fault model

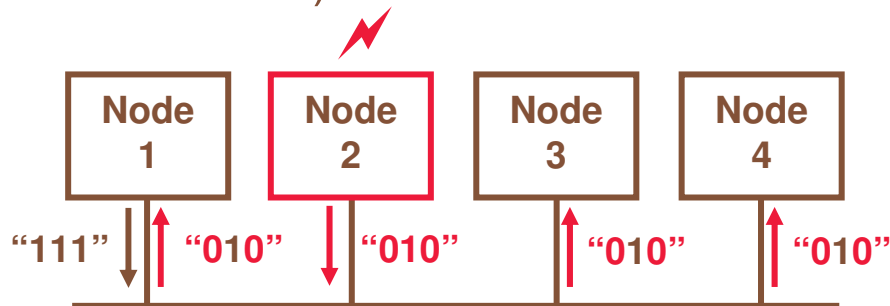
- *Stuck-at-dominant* fault (node or medium).



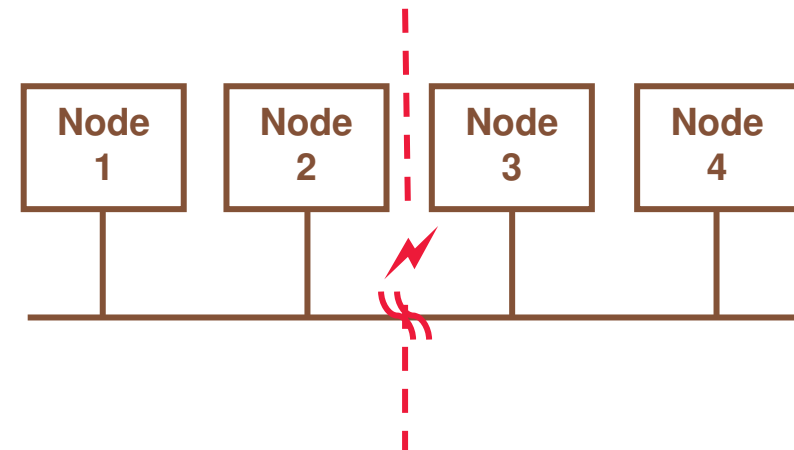
- *Stuck-at-recessive* fault (medium).



- *Bit-flipping* fault (node or medium).



- *Medium partition* fault.



# Introduction

The objective

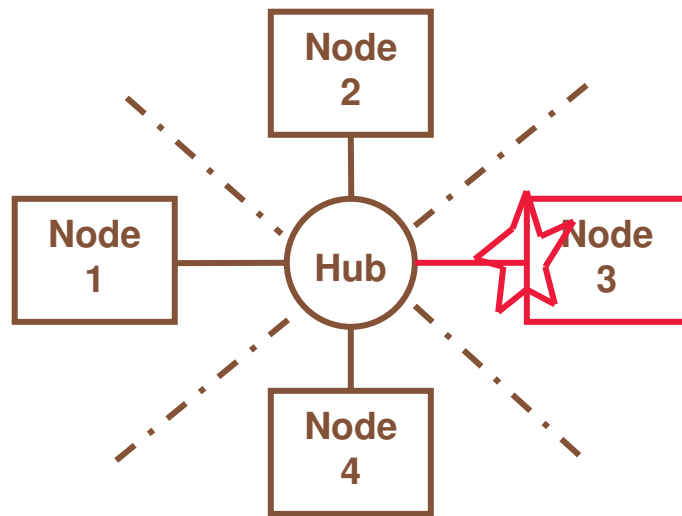
---

- To provide **communication infrastructures** that **improve error containment** and **reliability** of CAN.
- To **keep compatibility** with CAN: to **inherit its good properties** and to use **CAN-COTS** hardware and software.

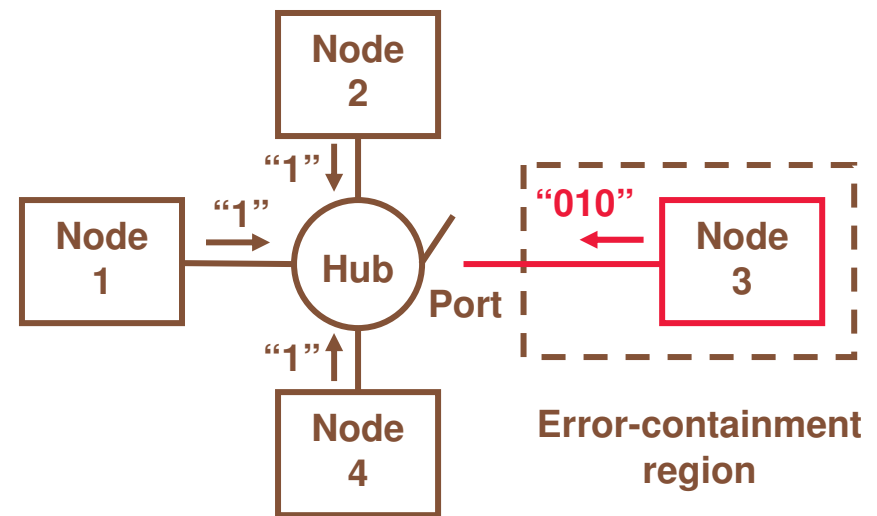
# Introduction

The solution: adequate star topologies

---



**no medium partitions**  
**no spatial proximity failures**



**no common-mode failures**

# Introduction

The solution: adequate star topologies

---

- An **adequate star** topology must provide.
  - ✓ **Error containment** of stuck-at and bit-flipping faults.
  - ✓ **Tolerance** of stuck-at and bit-flipping faults.
  - ✓ Full **compatibility** with CAN.

# Introduction

The solution: adequate star topologies



---

- An **adequate star** topology must provide.
  - ✓ **Error containment** of stuck-at and bit-flipping faults.
  - ✓ **Tolerance** of stuck-at and bit-flipping faults.
  - ✓ Full **compatibility** with CAN.

**This is what we have done**

# Outline

---

- CANcentrate.  **Error containment**
- ReCANcentrate.  **Error containment  
and reliability**
- Conclusions.
- Future work.

# CANcentrate

Main objective: error containment

---

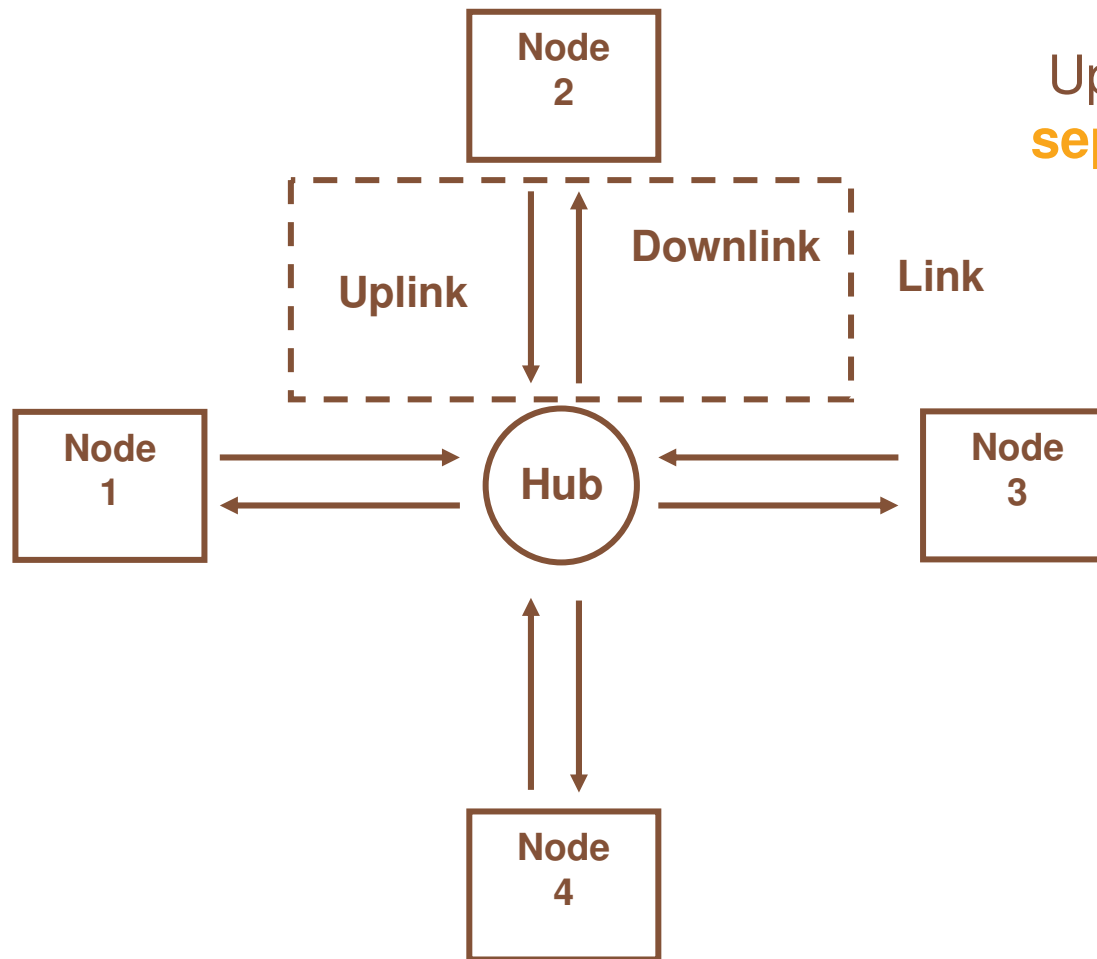
- To prevent that a single fault in a network component causes a severe failure of communication in a CAN network.
  - ✓ One fault just prevents a maximum of one node from communicating.



# CANcentrate

## Architecture overview

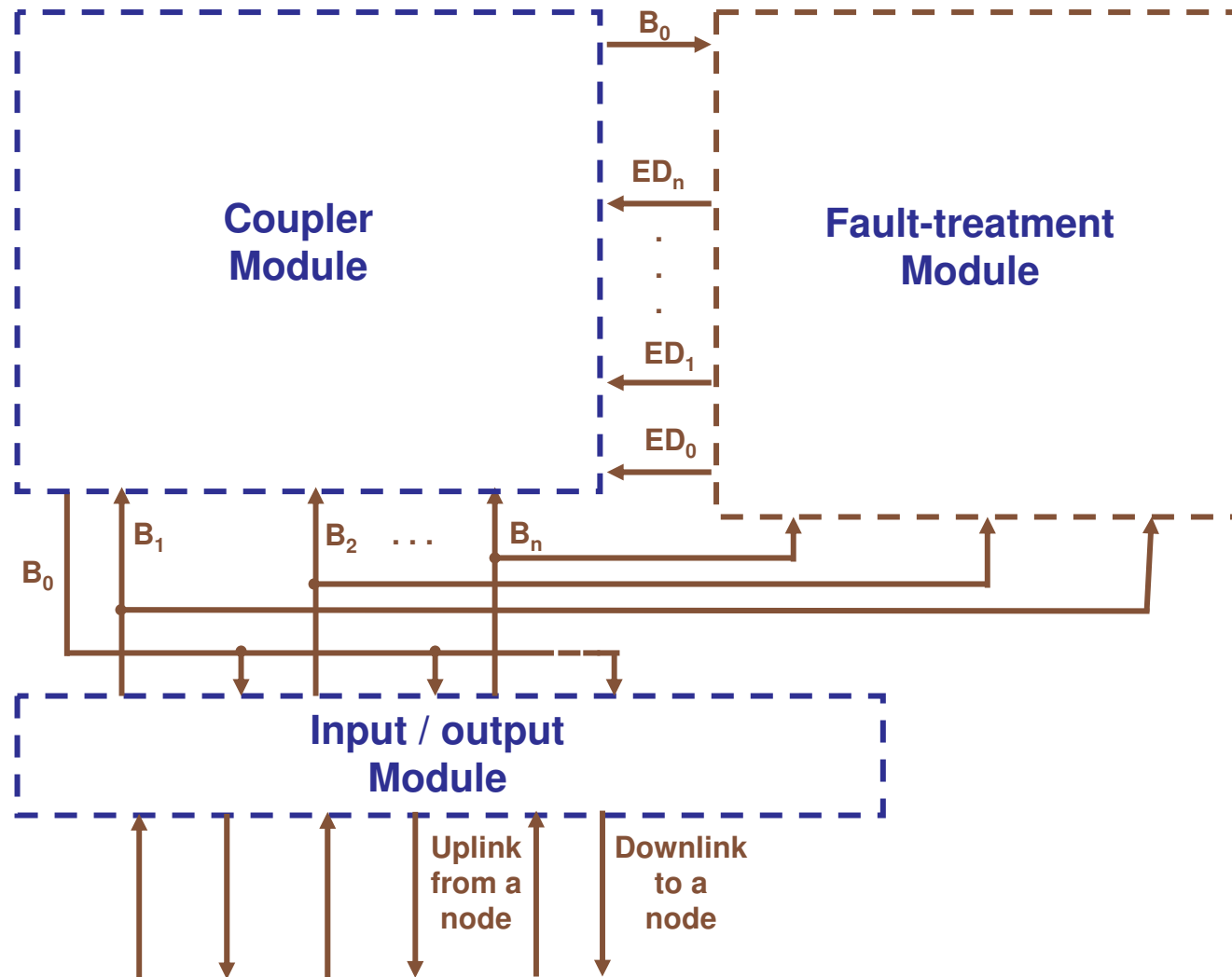
---



Uplink/downlink to **allow separating** the **contribution** of each hub port.

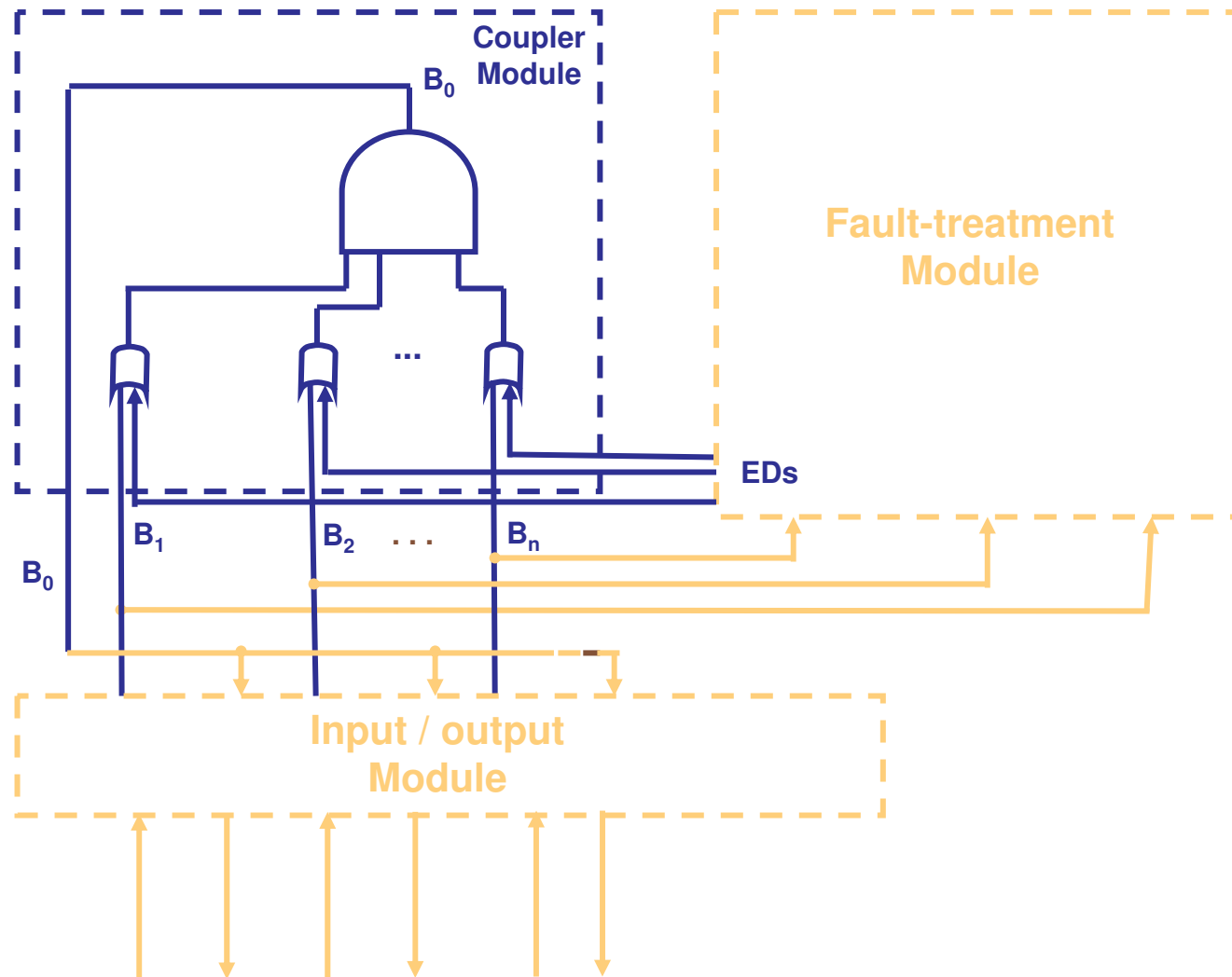
# CANcentrate

## Hub basic architecture



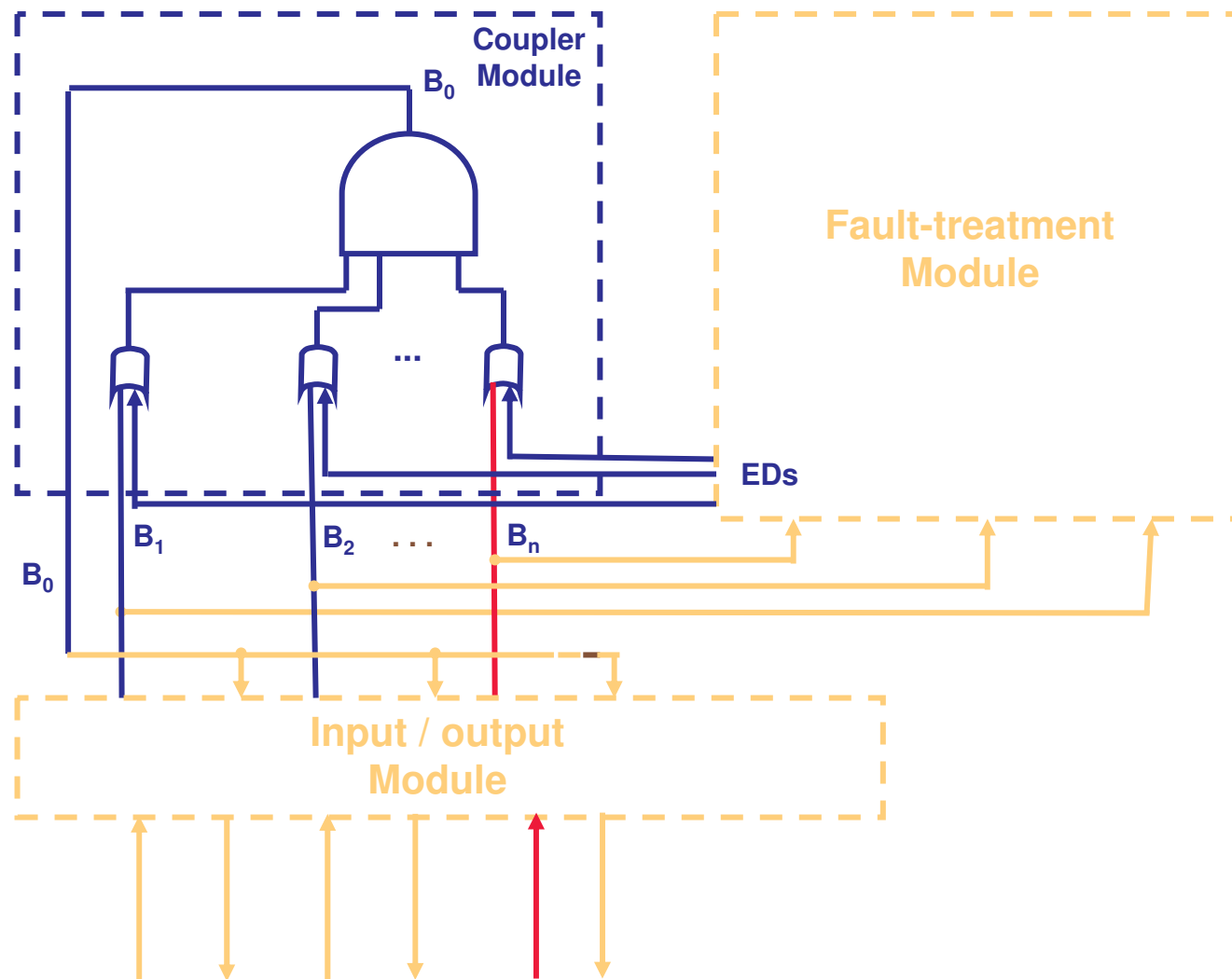
# CANcentrate

## Coupling schema



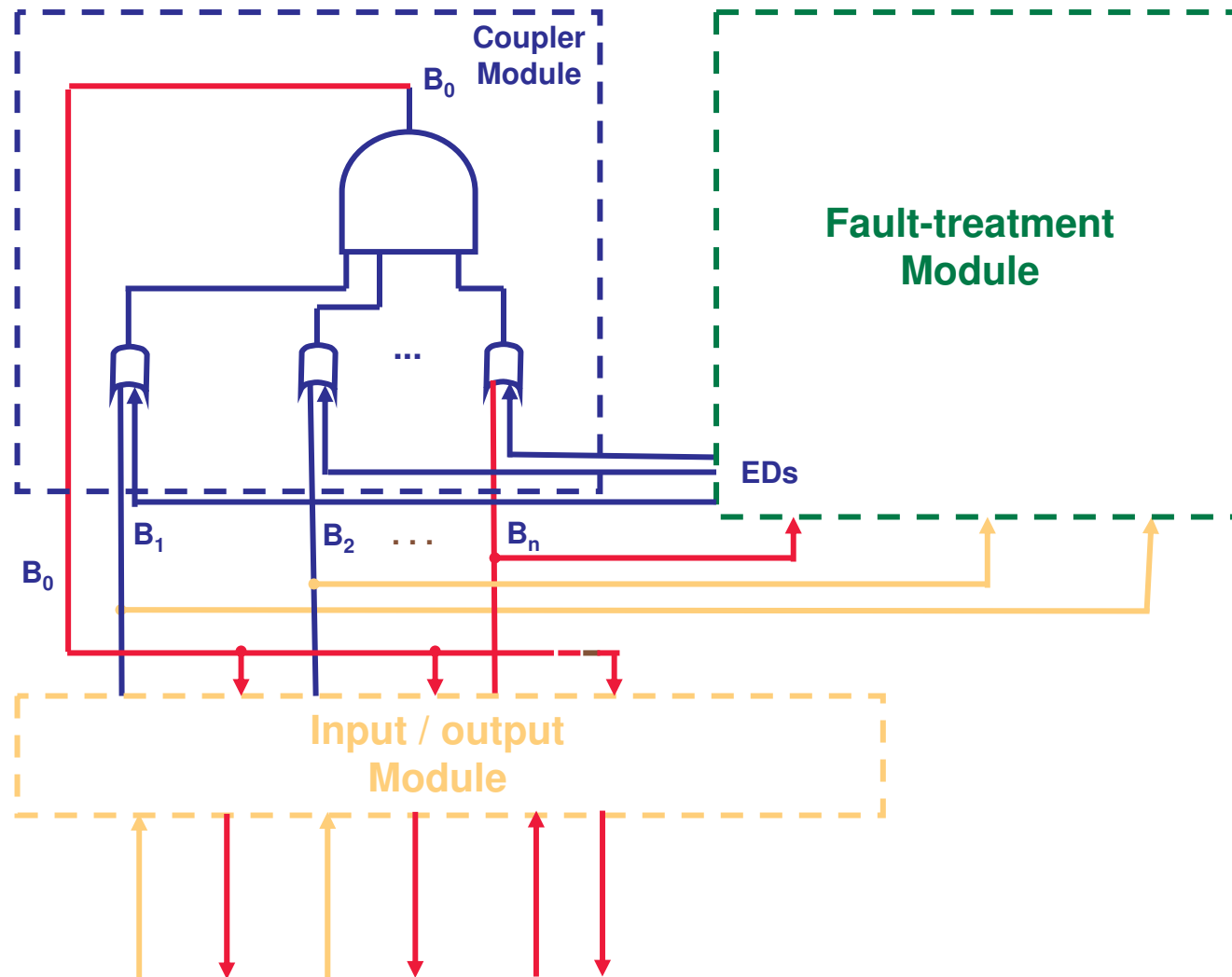
# CANcentrate

## Fault treatment basics



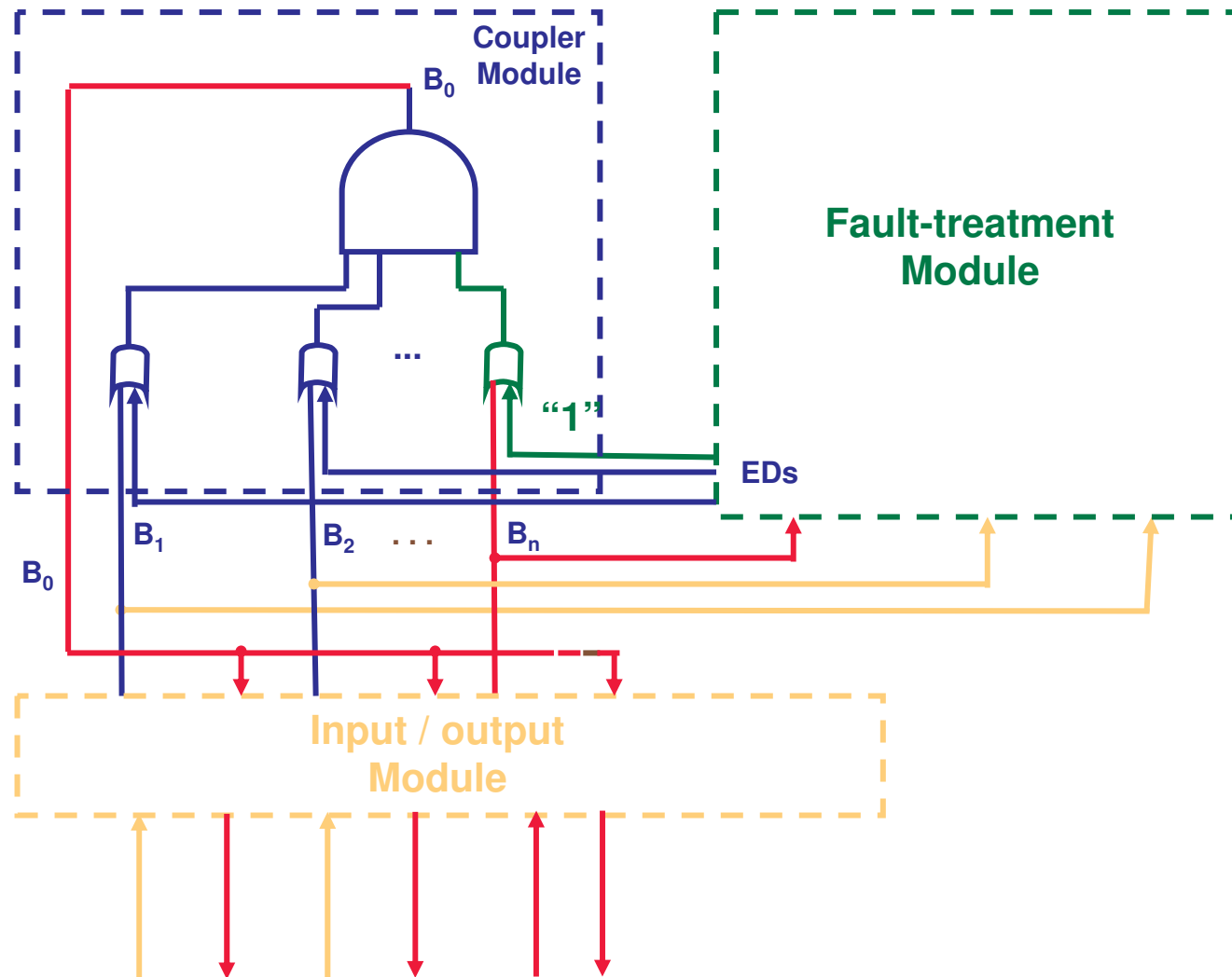
# CANcentrate

## Fault treatment basics



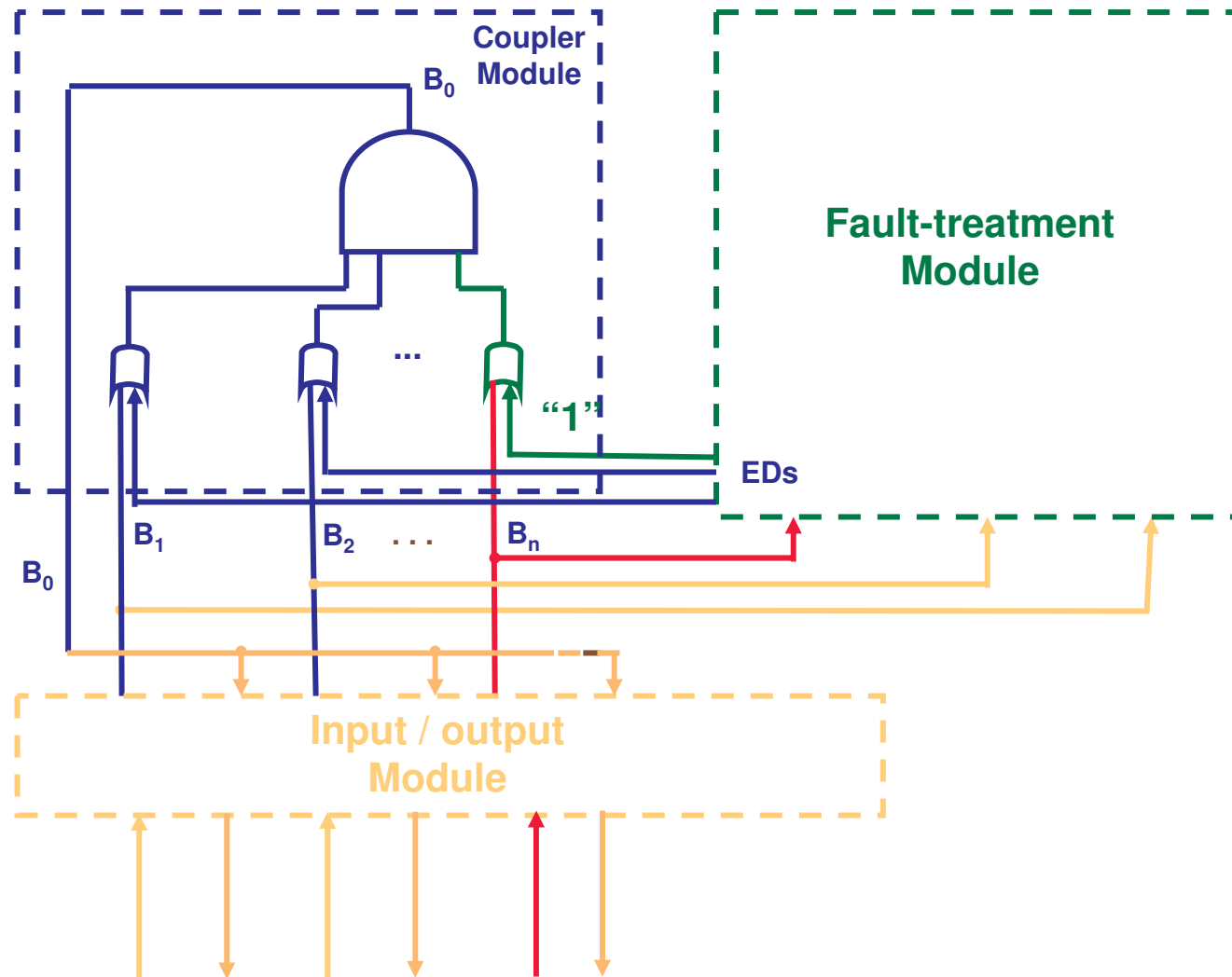
# CANcentrate

## Fault treatment basics



# CANcentrate

## Fault treatment basics



# CANcentrate

## Prototype implementation

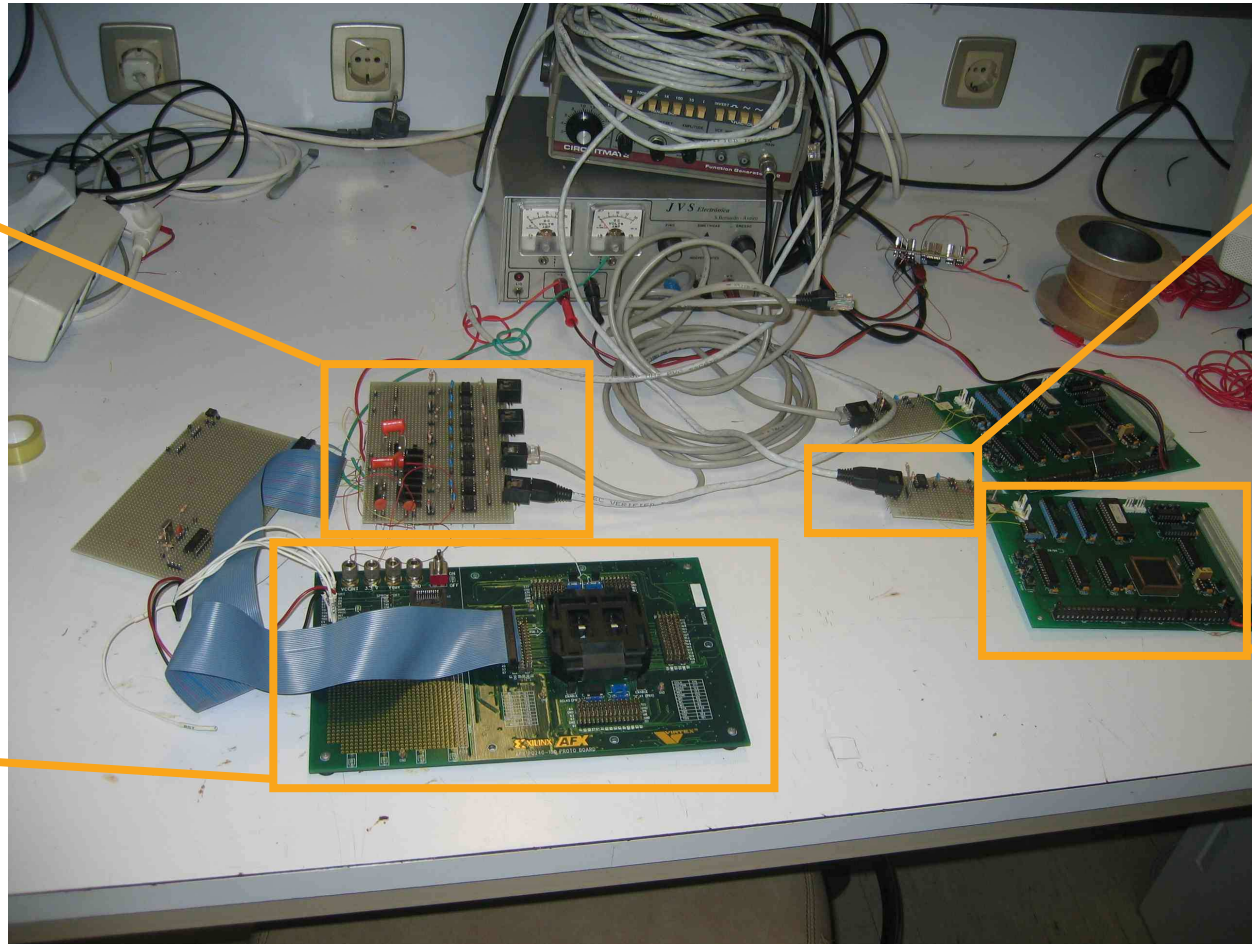
---

Input/Output  
Module

StarLink  
board

Hub core

CANivete  
board





# CANcentrate

## Prototype implementation - Tests

---

- Functional tests.

- ✓ **Short** fault isolation **delays** [25, 300]us at 690 kbs.

- Performance tests.

- ✓ **Inverse relationship** in CAN between the **bit rate** and the **network length**: at **690 kbs** the achieved a star diameter was **41 meters** (68 meter in CAN).
- ✓ **Extra delay** introduced by the hub **transceivers**. It does not visibly depend on the number of ports.

# CANcentrate

## Dependability evaluation

---

- A star **includes more hardware** than a bus: the **probability** of suffering from a **fault** is **higher** in a star.
  - ✓ CANcentrate **reduces reliability**.
  - ✓ But CANcentrate can **improve error containment**.
    - ☺ Suitable for system that can **assume that up to K of N nodes** cannot communicate.

# CANcentrate

Dependability evaluation – Modelling framework

---

- Dependability **comparison** in the presence of **permanent** hardware faults.
- **CAN** and **CANcentrate** modelled by means of: **Stochastic Activity Networks** (SANs): a generalization of Stochastic Petri Nets.
- **Realistic** values for **dependability parameters** such as failure rates and error-detection coverages.

# CANcentrate

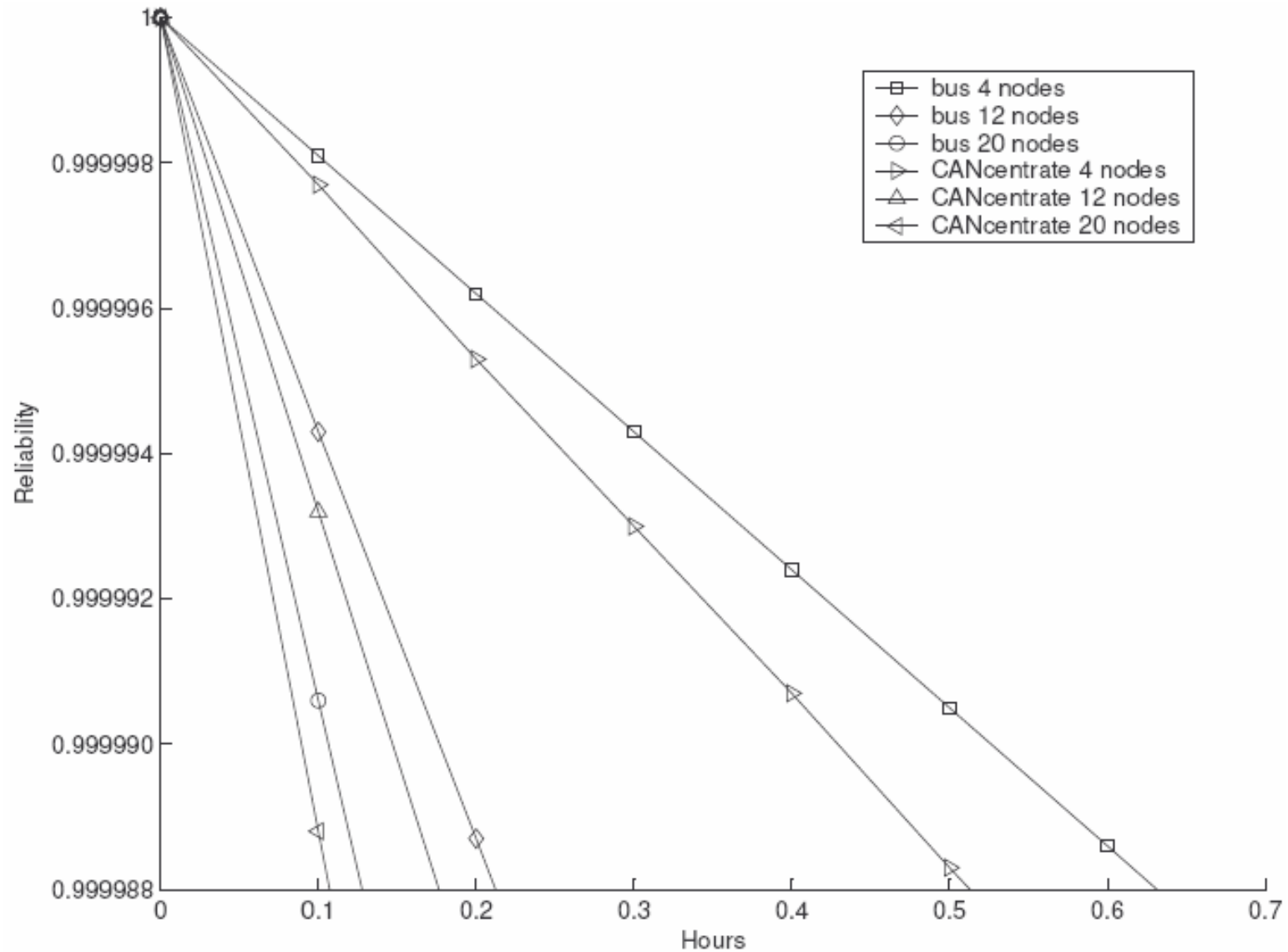
## Dependability evaluation – Assumptions

---

- **Results** are **lower bounds** to the dependability of CANcentrate.
  - ✓ Modeling assumptions that **favor CAN**, e.g. we did not consider spatial proximity failures.

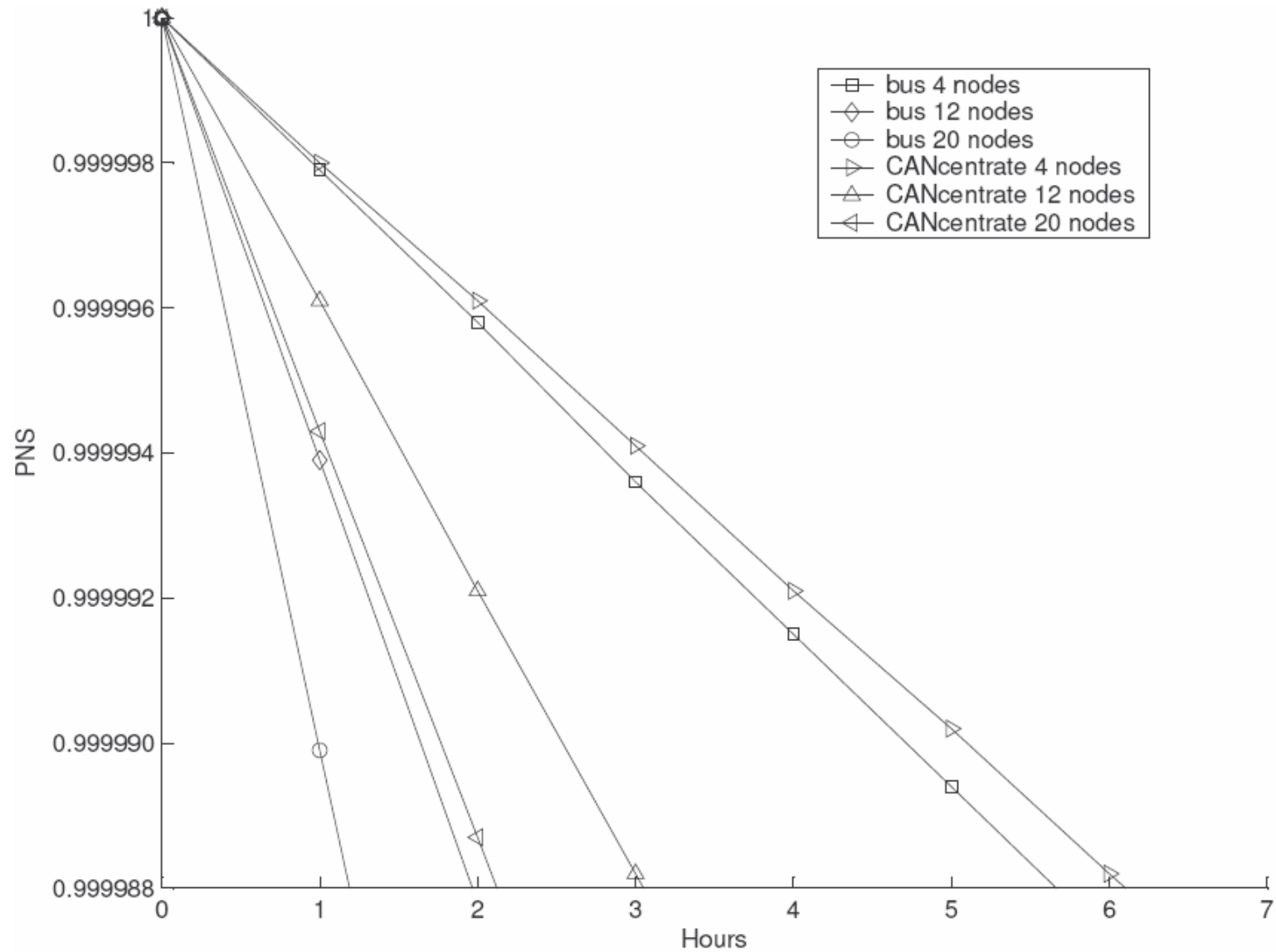
# CANcentrate

## Reliability comparison vs number of nodes



# CANcentrate

## PNS comparison vs number of nodes

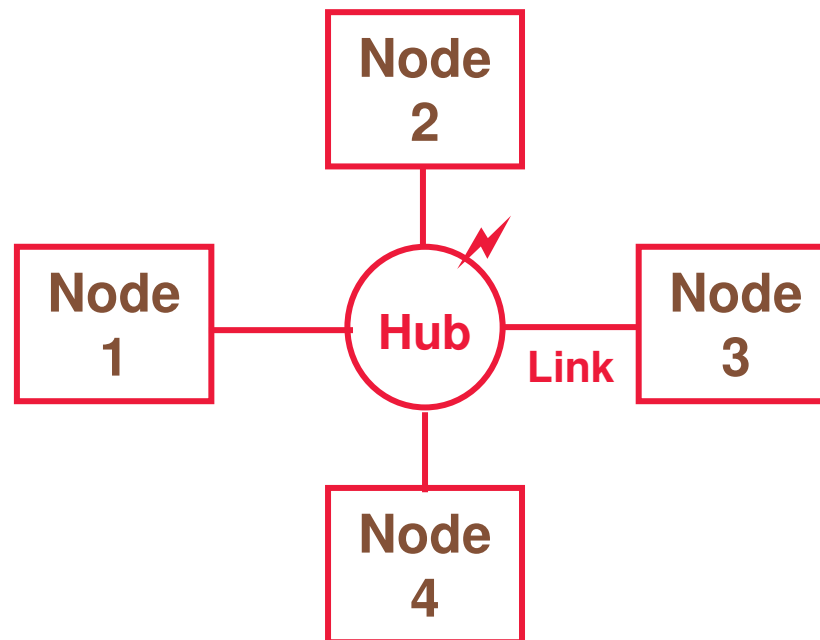


# CANcentrate

## Main disadvantages


---

- CANcentrate slightly reduces the reliability.
- It still has one severe point of failure: the hub.



# Outline

---

- CANcentrate.
- **ReCANcentrate.**  **Error containment and reliability**
- Conclusions.
- Future work.



# ReCANcentrate

Main objectives: error containment and reliability

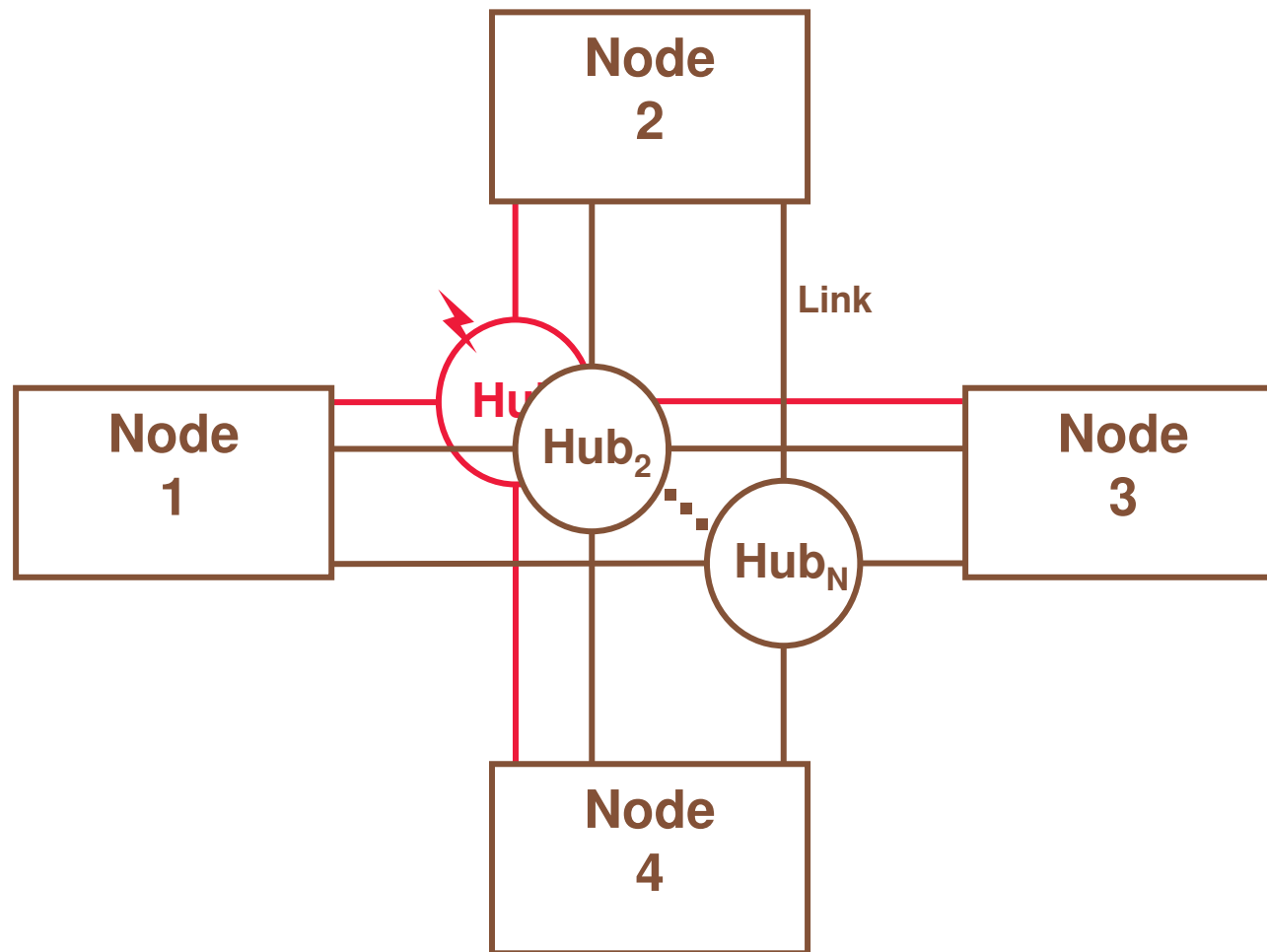
---

- To **definitively eliminate all points** of severe failure in a CAN network: **tolerate one hub failure**.
- To tolerate link failures.

# ReCANcentrate

The solution: a replicated star

---



# ReCANcentrate

A replication of CANcentrate

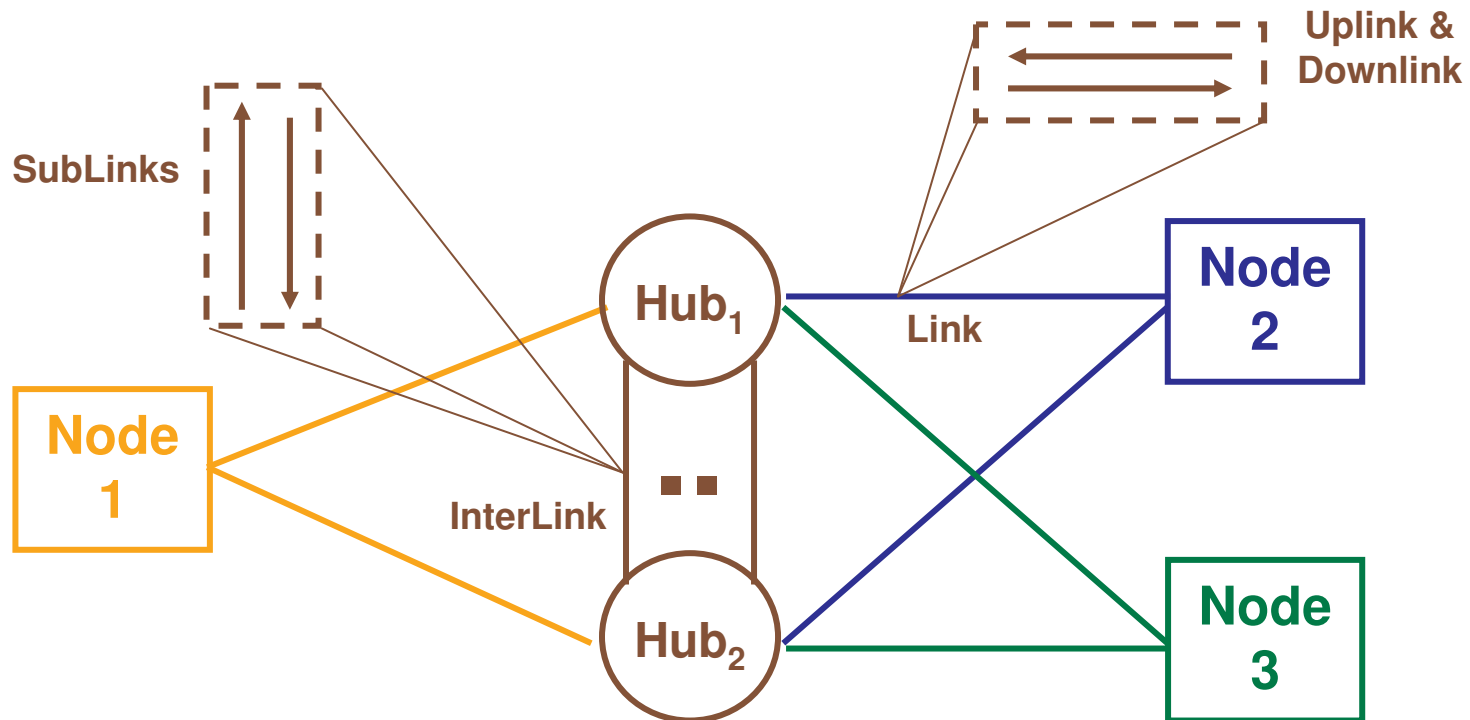
---

- In particular: **we replicated CANcentrate.**
  - ✓ We take advantage of the error-containment properties already achieved by CANcentrate.
  - ✓ We still keep the fully compatibility with CAN.

# ReCANcentrate

## Architecture overview

- Two **coupled** hubs.

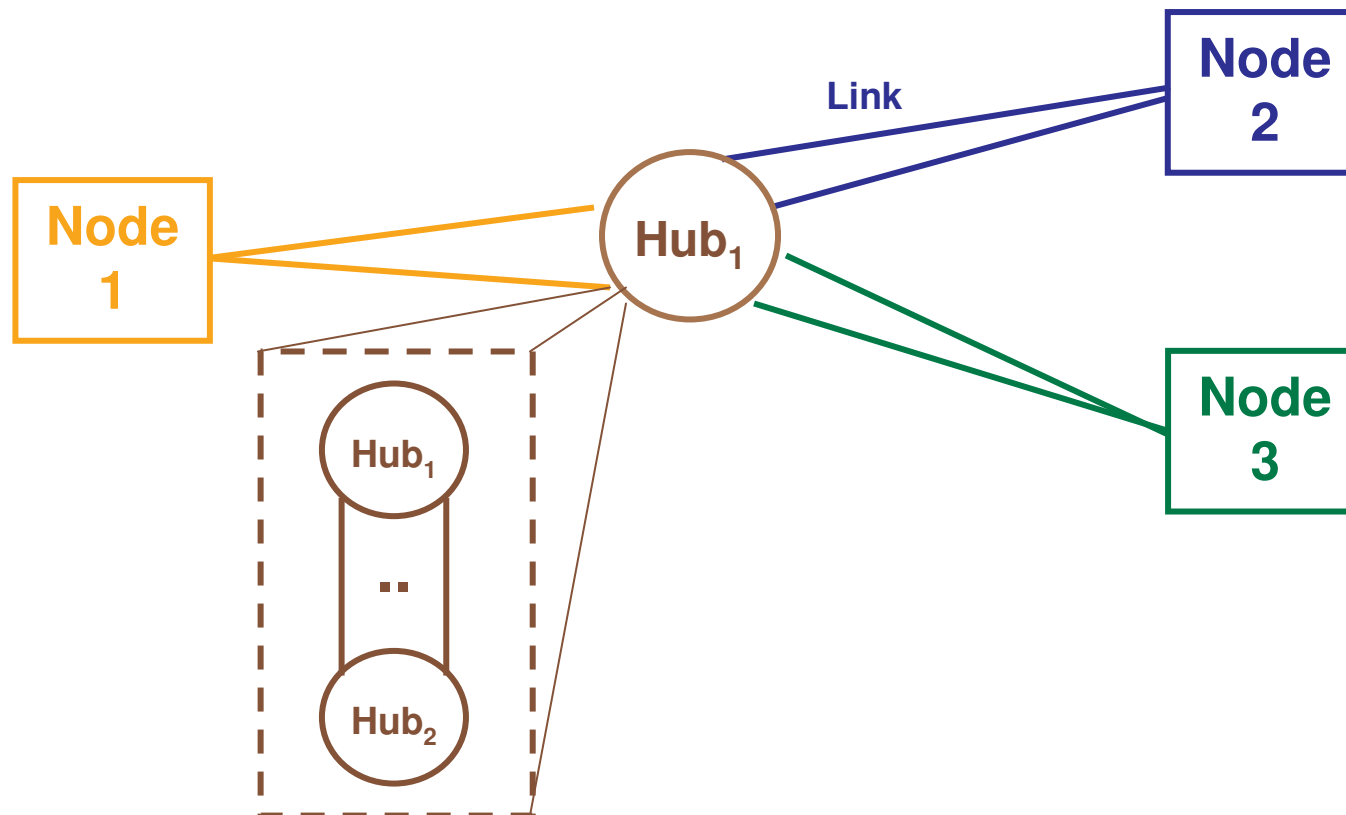


# ReCANcentrate

## Basic functionality

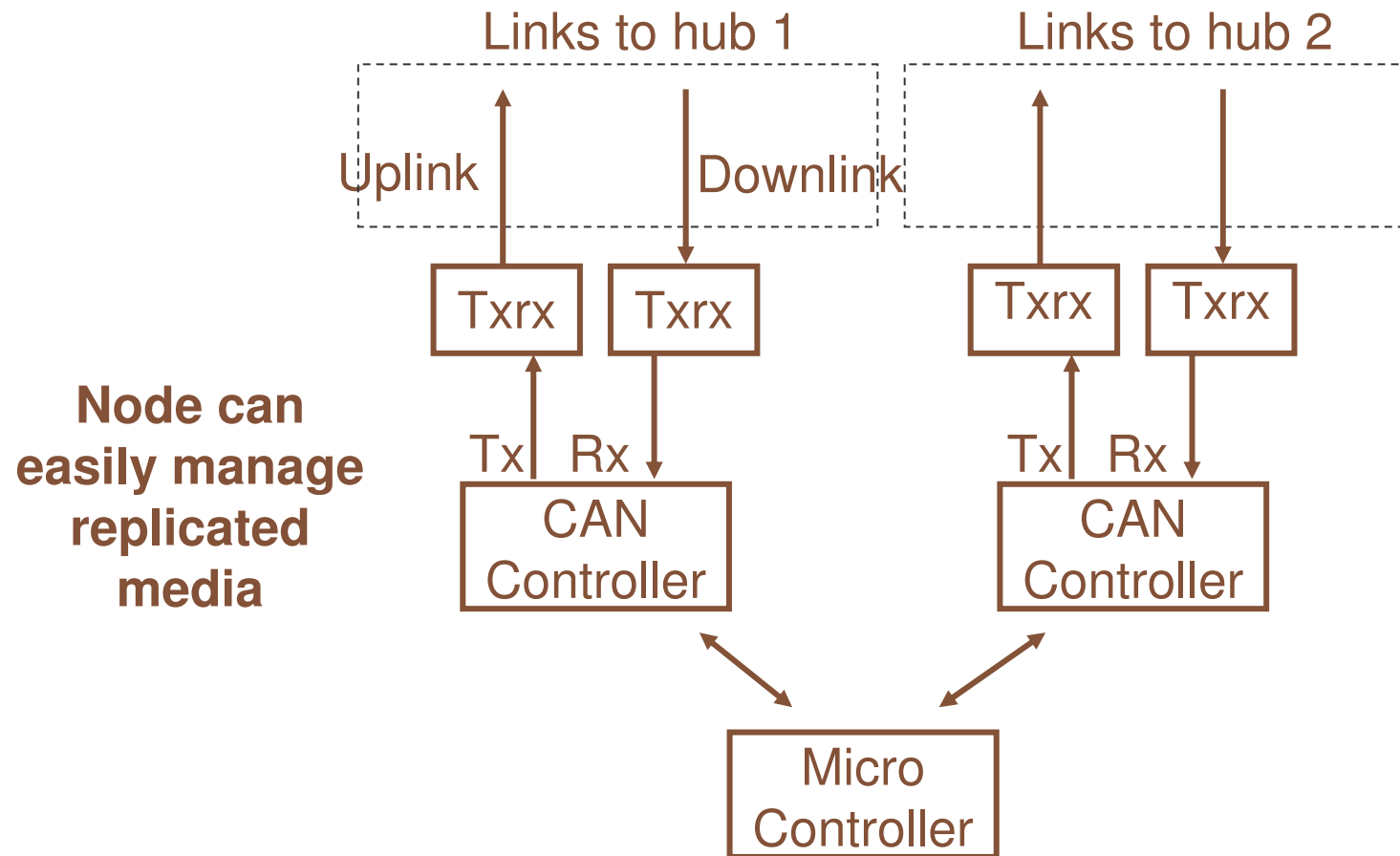
---

- Hubs behave like one: they send the same bit stream bit by bit to the nodes.



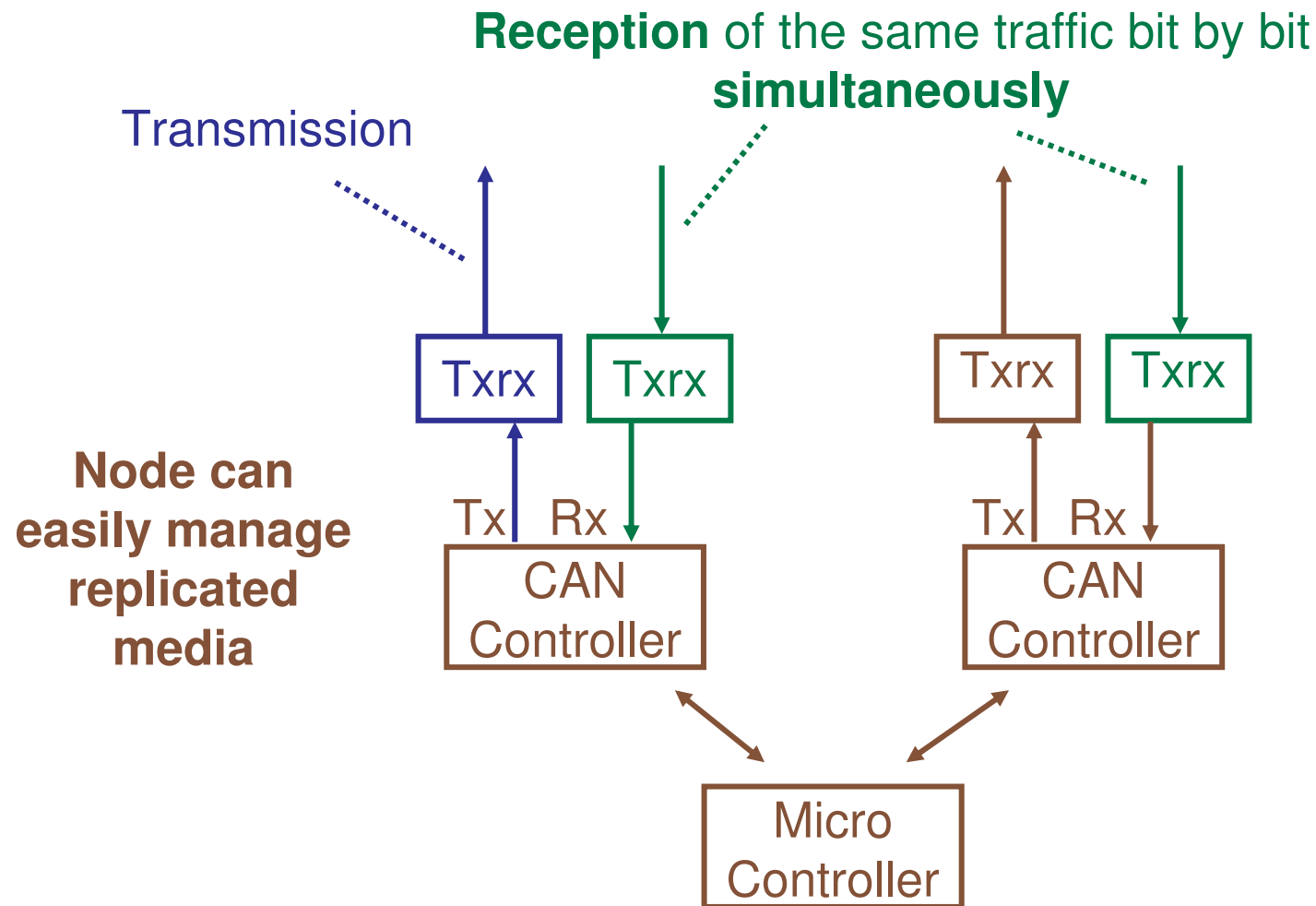
# ReCANcentrate

## Basic functionality



# ReCANcentrate

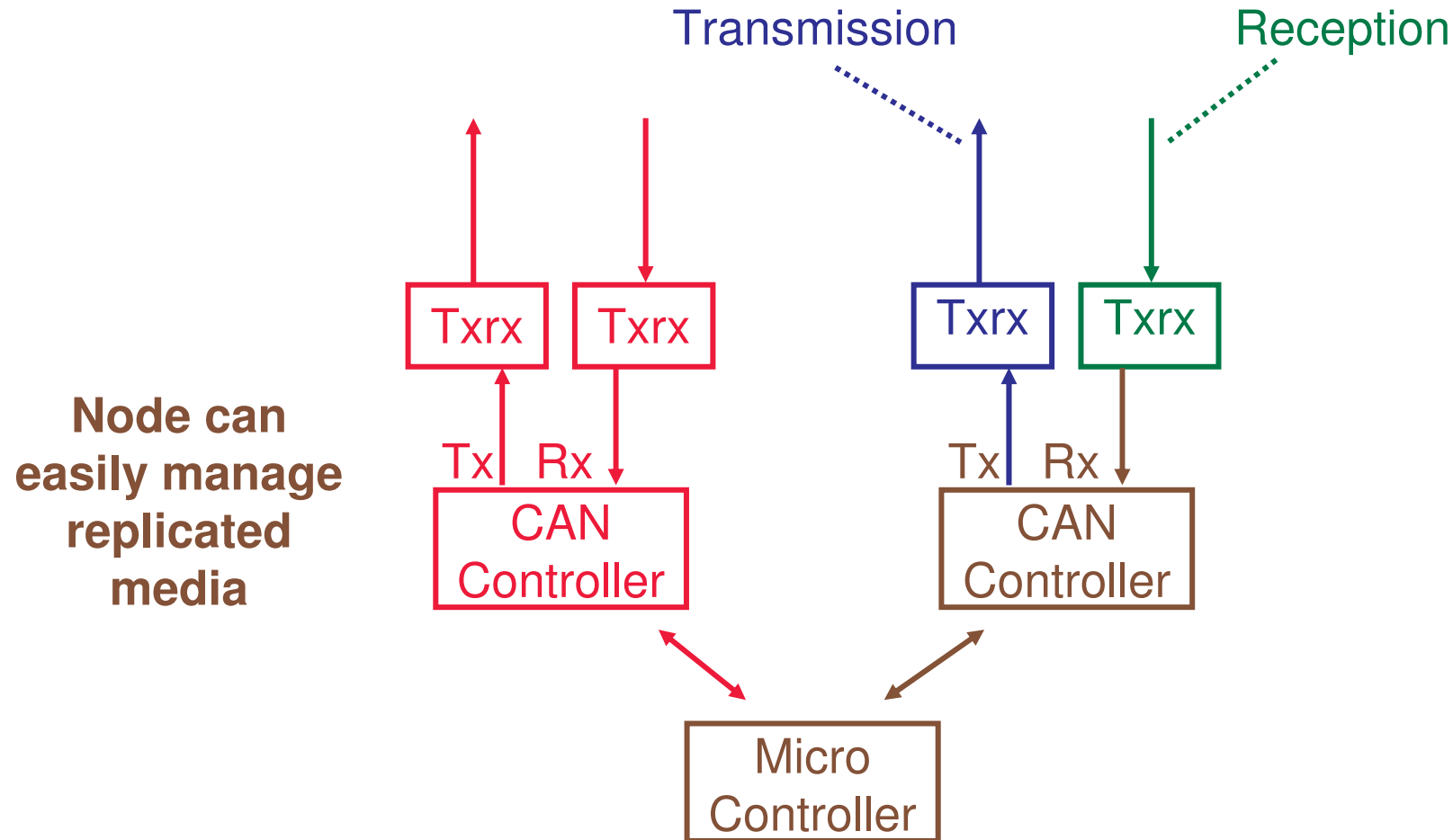
## Basic functionality



# ReCANcentrate

## Basic functionality

---



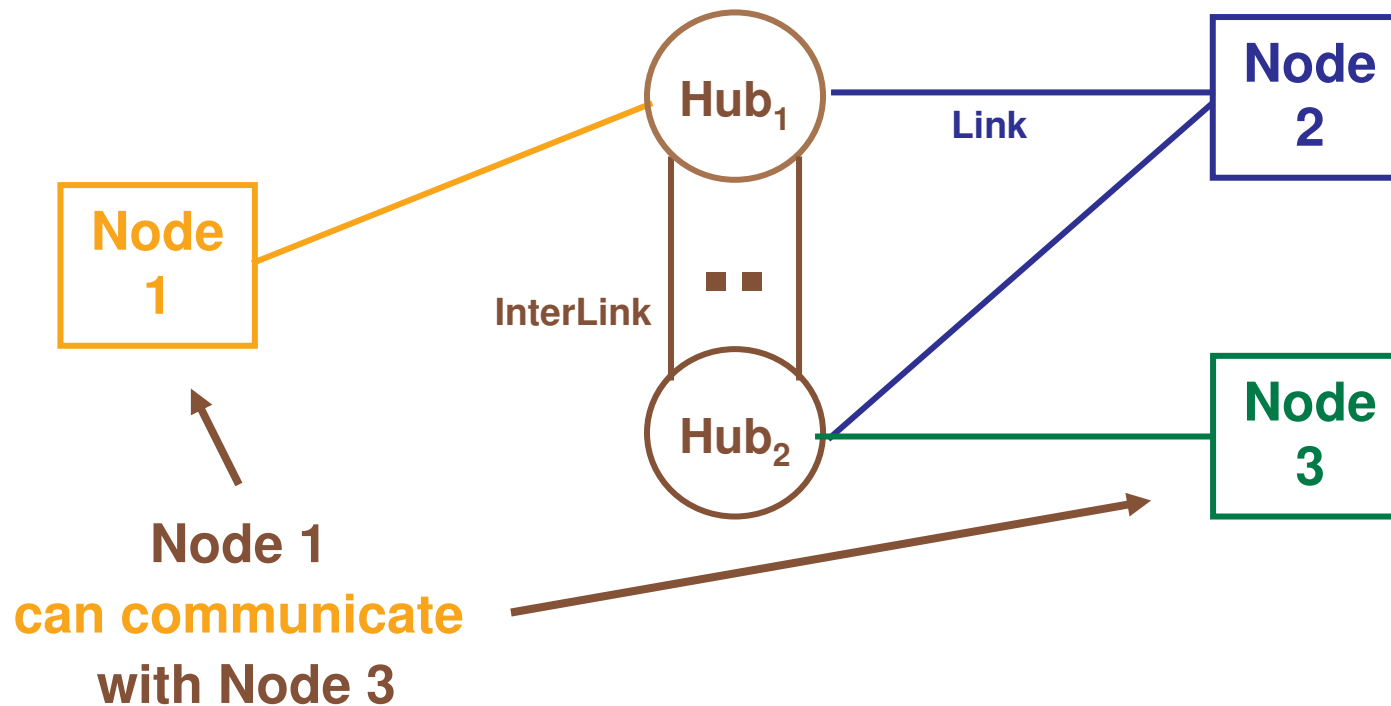


# ReCANcentrate

## Basic functionality

---

- **Flexible** configuration to **reduce cabling** costs.

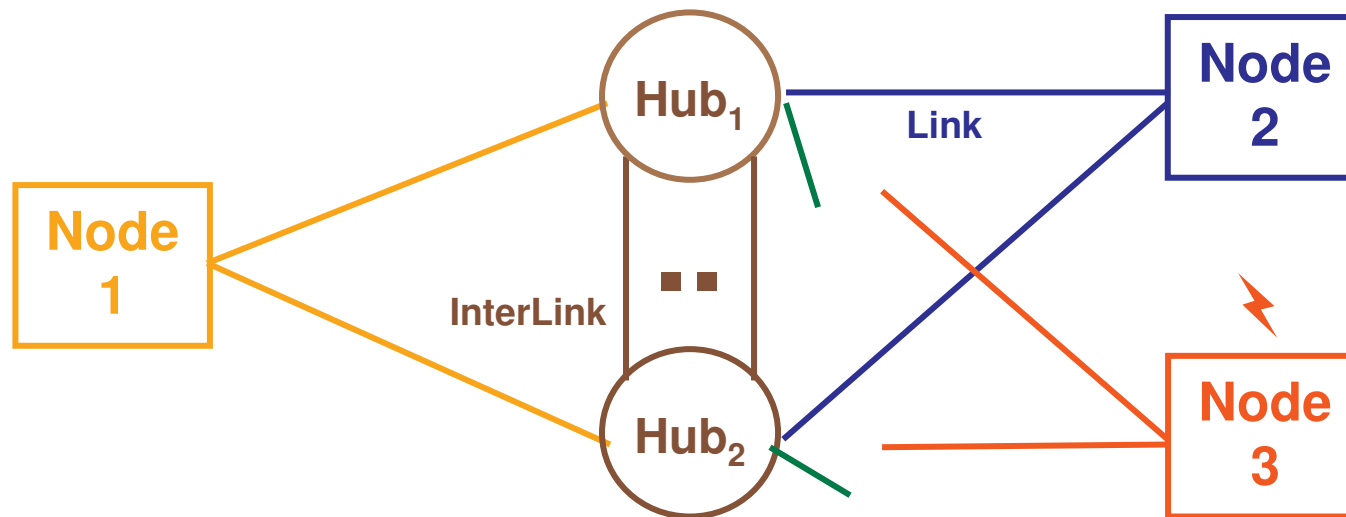


# ReCANcentrate

## Basic functionality

---

- Error containment of **link** and **node faults**.

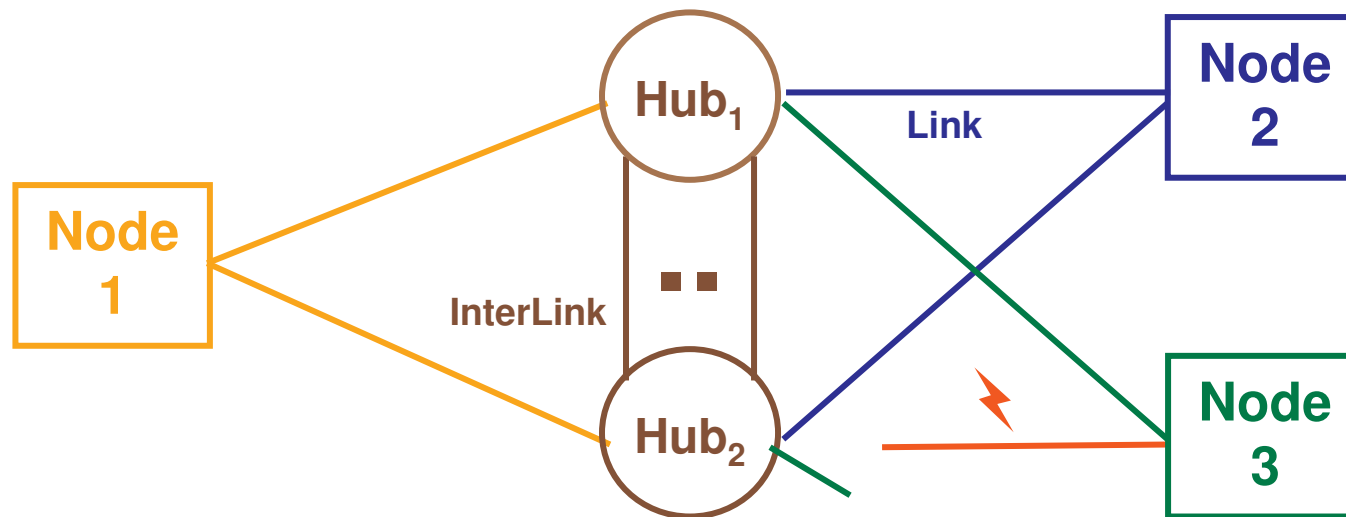


# ReCANcentrate

## Basic functionality

---

- **Tolerance** to **link** faults.

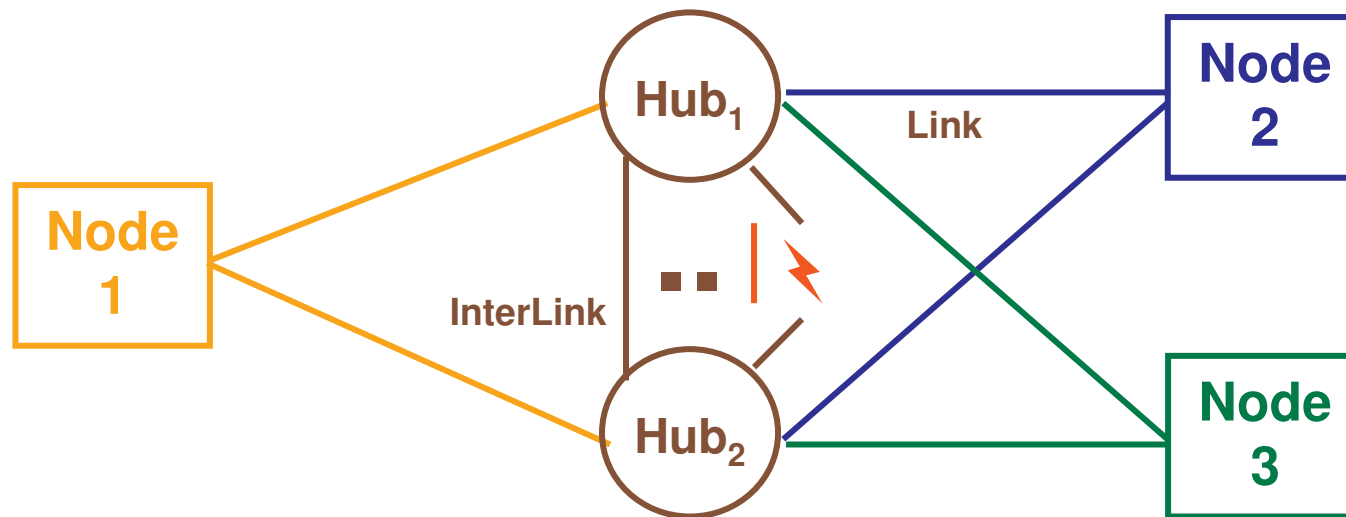


# ReCANcentrate

## Basic functionality

---

- **Tolerance** to **interlink** faults.

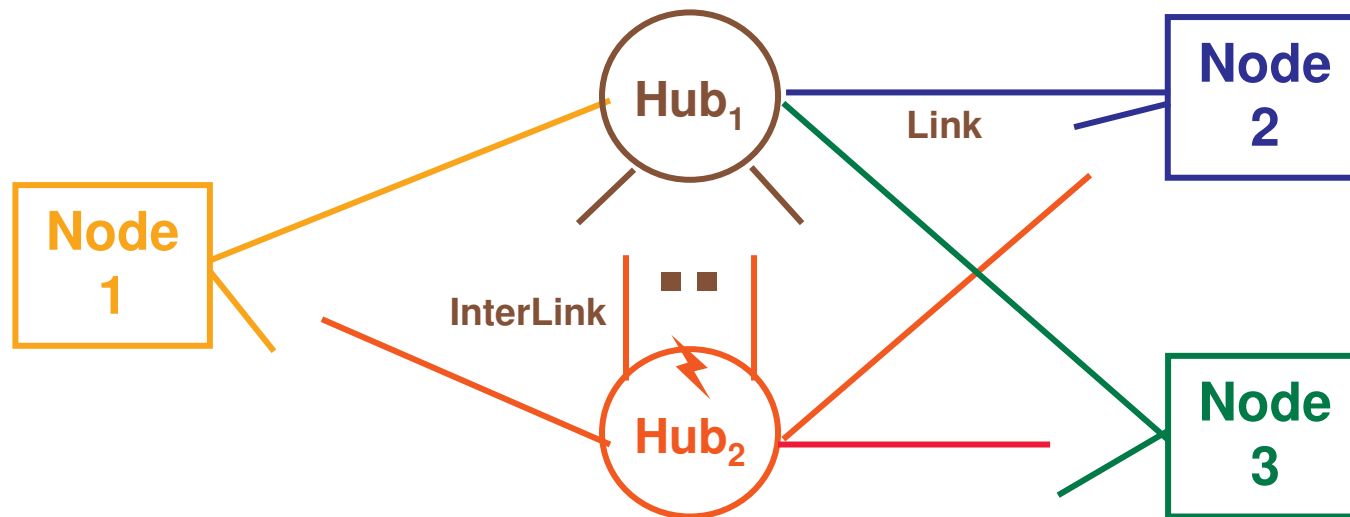


# ReCANcentrate

## Basic functionality

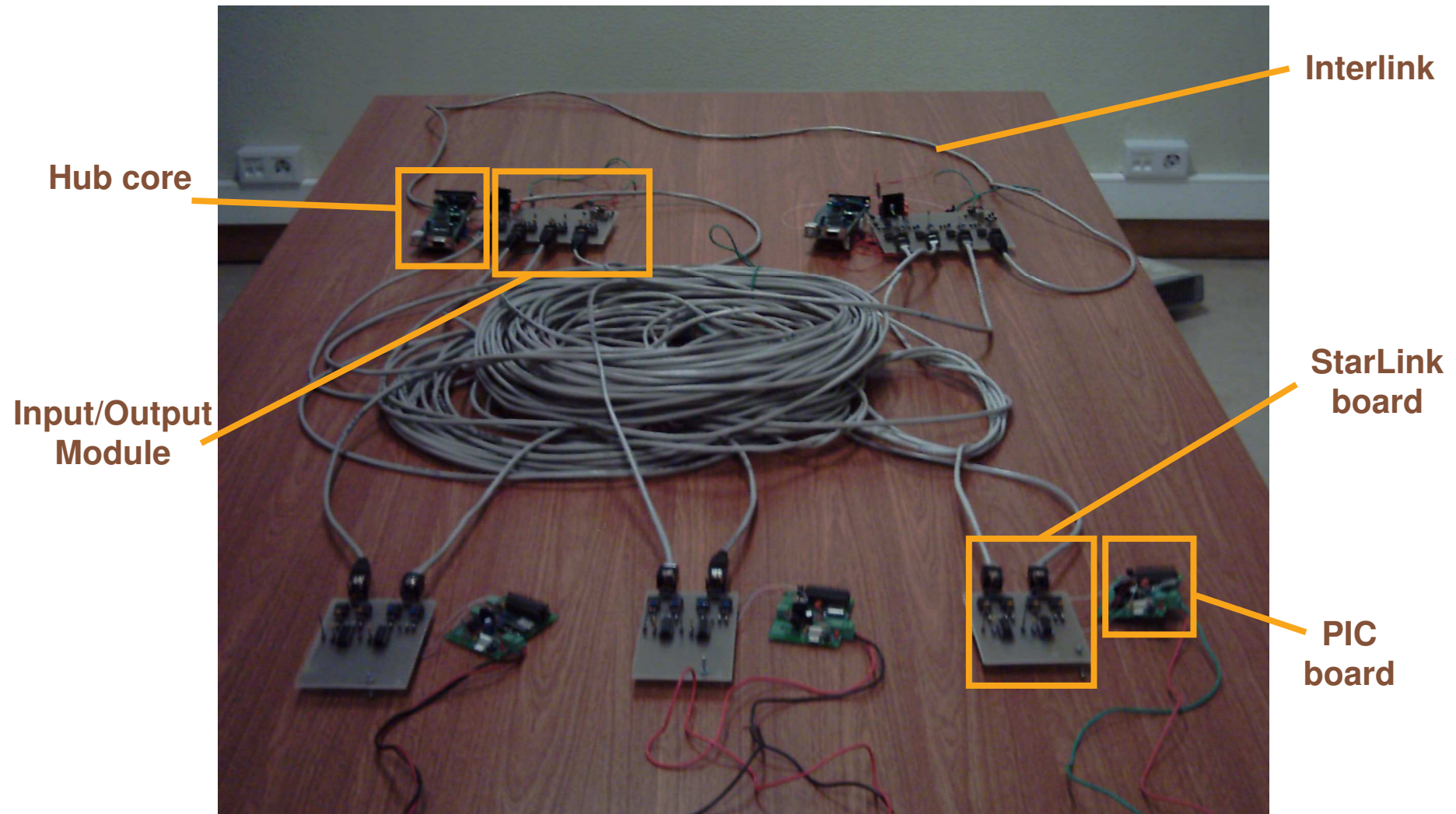
---

- **Tolerance** to **hub** faults.



# ReCANcentrate

Prototype implementation



# ReCANcentrate

## Prototype implementation - Tests

---

- Functional tests.
  - ✓ **Similar results** as in CANcentrate.
- Performance tests.
  - ✓ At **625 kbs**, the maximum achievable star diameter was **25 meters** (79 meters in CAN).

# ReCANcentrate

Dependability evaluation

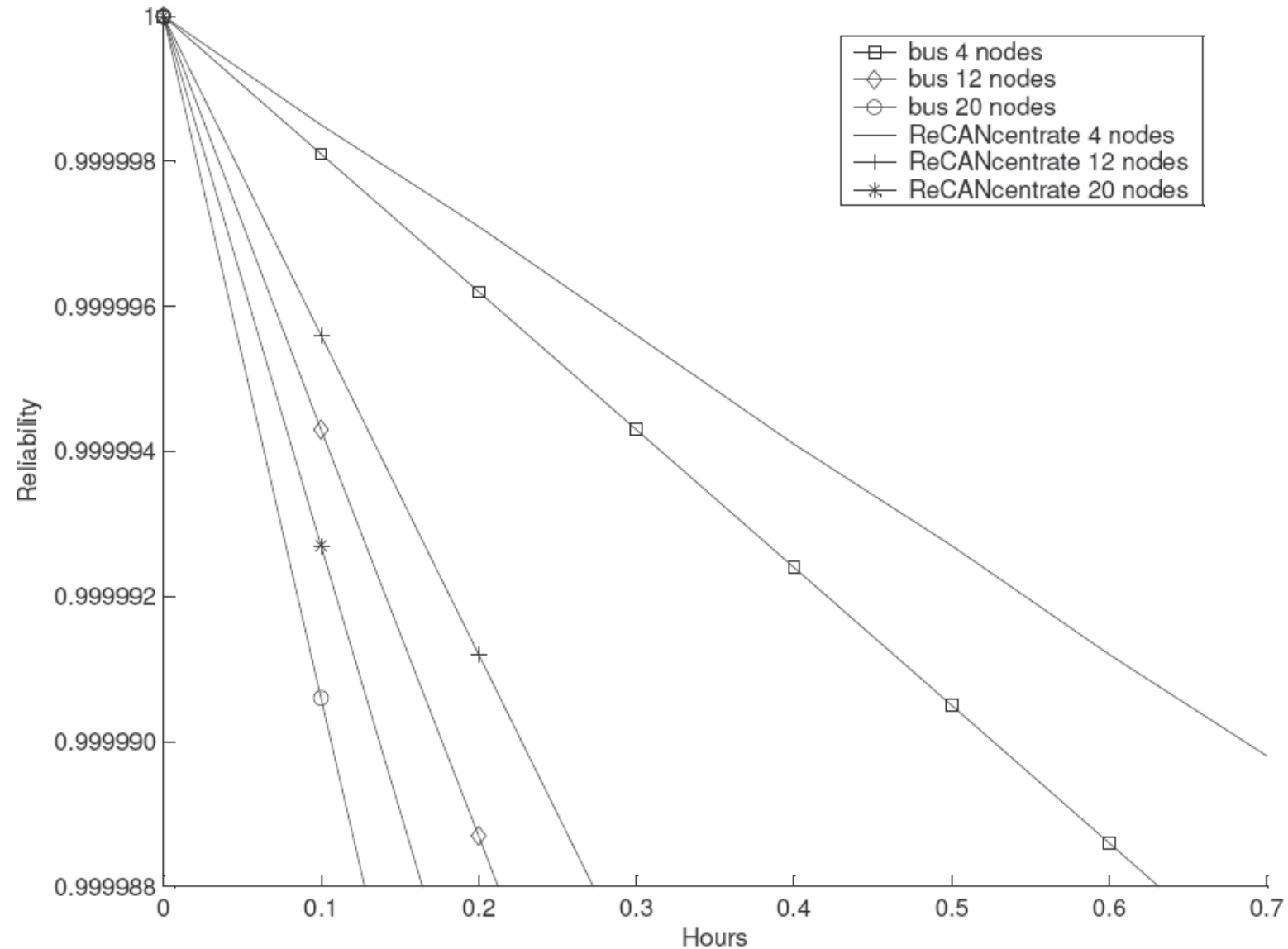
---

- ReCANcentrate modeled using the **same formalisms** and tools as for CANcentrate.
- **Results** are **lower bounds** to the dependability of ReCANcentrate.



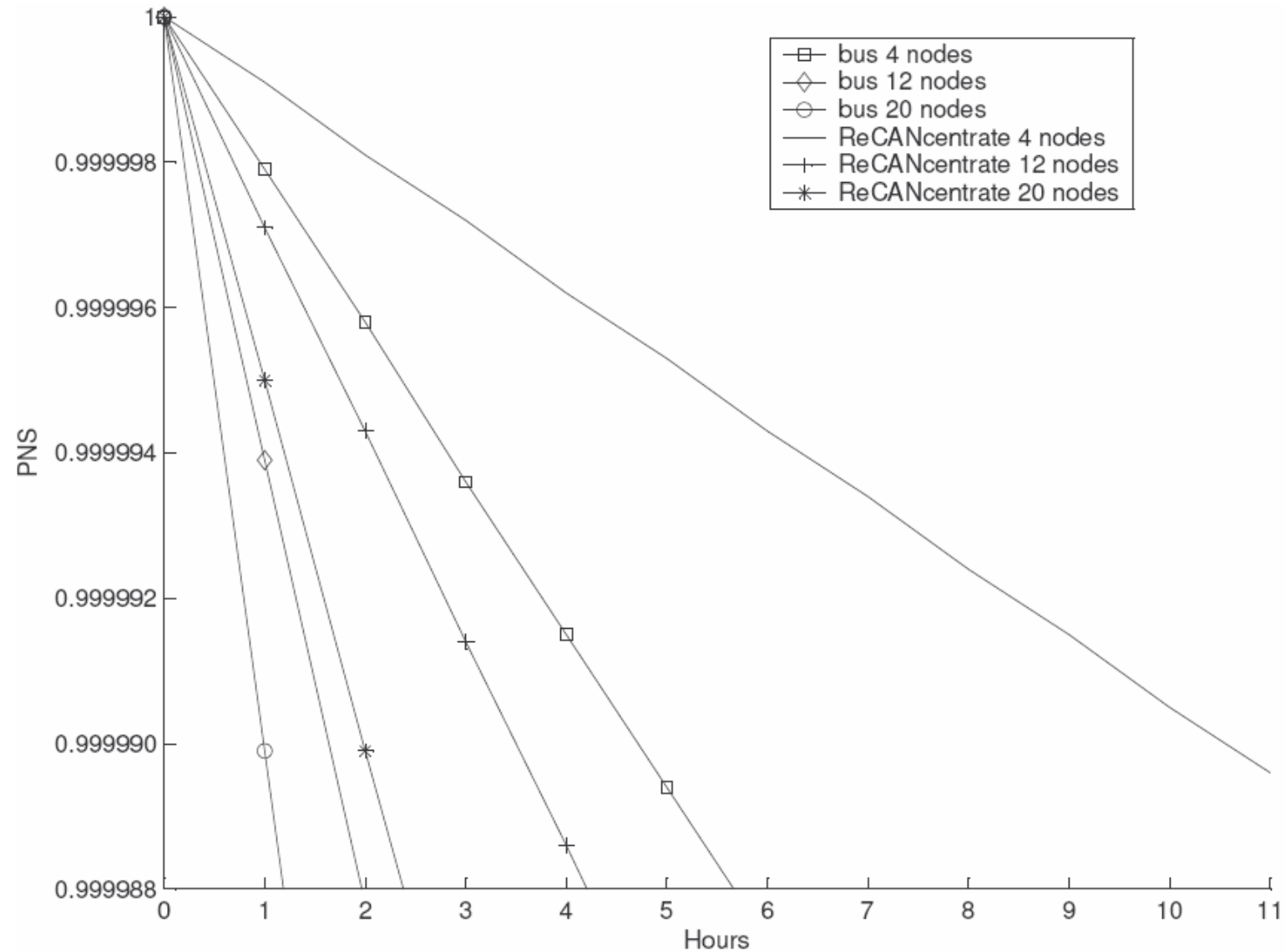
# ReCANcentrate

## Reliability comparison vs number of nodes



# ReCANcentrate

## PNS comparison vs number of nodes



# Conclusions

---

- CANcentrate **demonstrates** that it is possible to **improve error containment** of CAN by means of a CAN-compliant simplex star topology.
- ReCANcentrate **demonstrates** that it is possible to **improve both reliability and error containment** of CAN by means of a replicated star topology.

# Future work

---

- Design and implementation of further fault treatment mechanisms at hubs: babbling idiot, masquerading faults, etc.
- Design and implementation of stars that use only one CAN cable per link.
- Performability evaluation of (Re)CANcentrate in the presence of transient faults.
- Implementation and formal verification of a driver for managing the replicated media in ReCANcentrate.



---

# Improving Error Containment and Reliability of Controller Area Network (CAN) by means of Adequate Star Topologies

---

Manuel Barranco

Julián Proenza

Luis Almeida