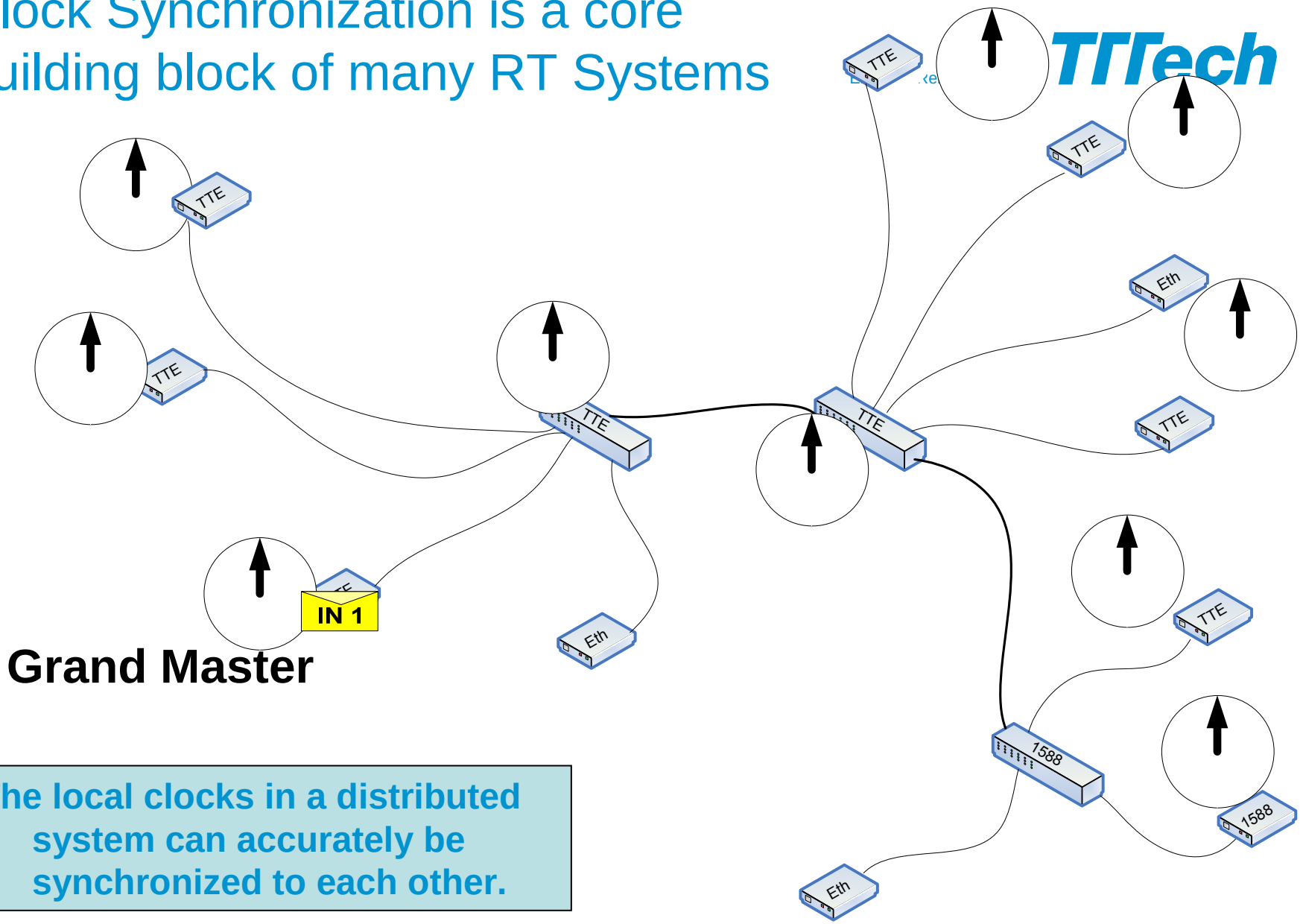
Discussion of Failure Mode Assumptions for IEEE 802.1Qbt

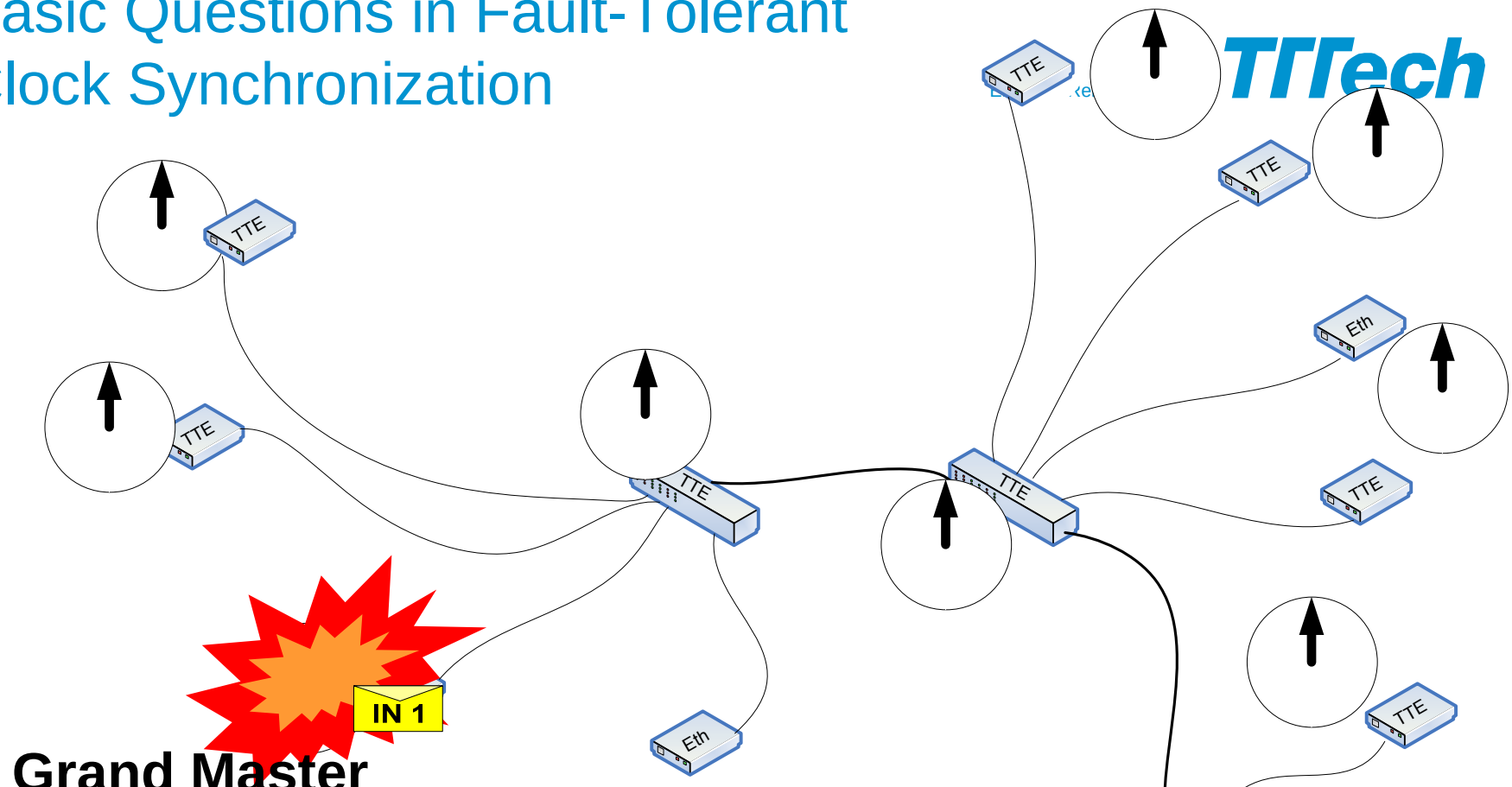
Wilfried Steiner, Corporate Scientist
wilfried.steiner@tttech.com

Clock Synchronization is a core building block of many RT Systems



The local clocks in a distributed system can accurately be synchronized to each other.

Basic Questions in Fault-Tolerant Clock Synchronization



Loss of Grand Master clock requires a changeover
- **How long does the changeover take?**
- **Is the changeover fault-tolerant?**
- **Is a malicious failure behavior of the Grand Master clock tolerated?**

Fault-Tolerance through Redundancy

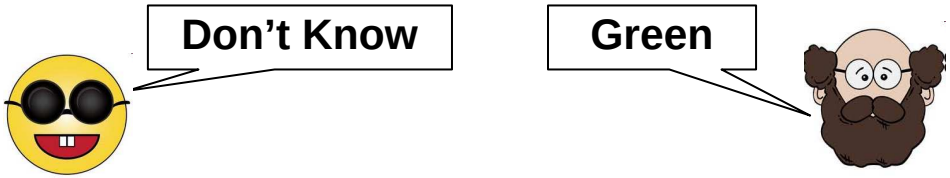
Ensuring Reliable Networks TTTech

Situation:

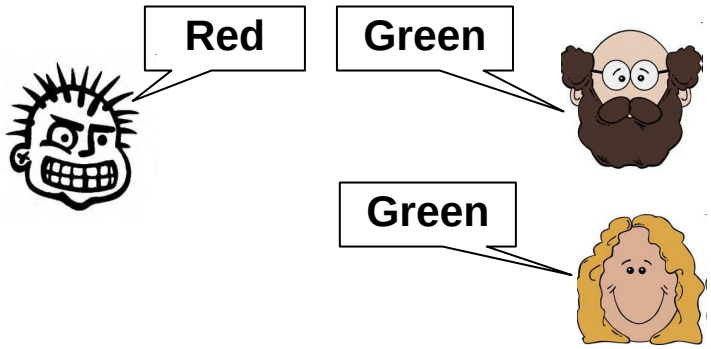
What is the color of the house?

No Failure

Fail-Silence Failure



Fail-Consistent Failure



Failure Mode: Fail-Silence

When the current grandmaster clock fails then gPTP ensures that another clock becomes the new grandmaster

- if there exists such a clock in the system, which we will assume in the following

This means that there is some fail-over time after which the system is running stable again – synchronized and syntonized to the new grandmaster clock.

The fail-silence failure mode is tolerated

- when the original grand master clock fails permanently.

Failure Mode: Fail-Silence

What happens when the original grandmaster clock fails transiently or intermittent?

- e.g., the original grandmaster clock periodically reboots

→Will the network oscillate between the original and a secondary grandmaster clock?

Model-Based Development i

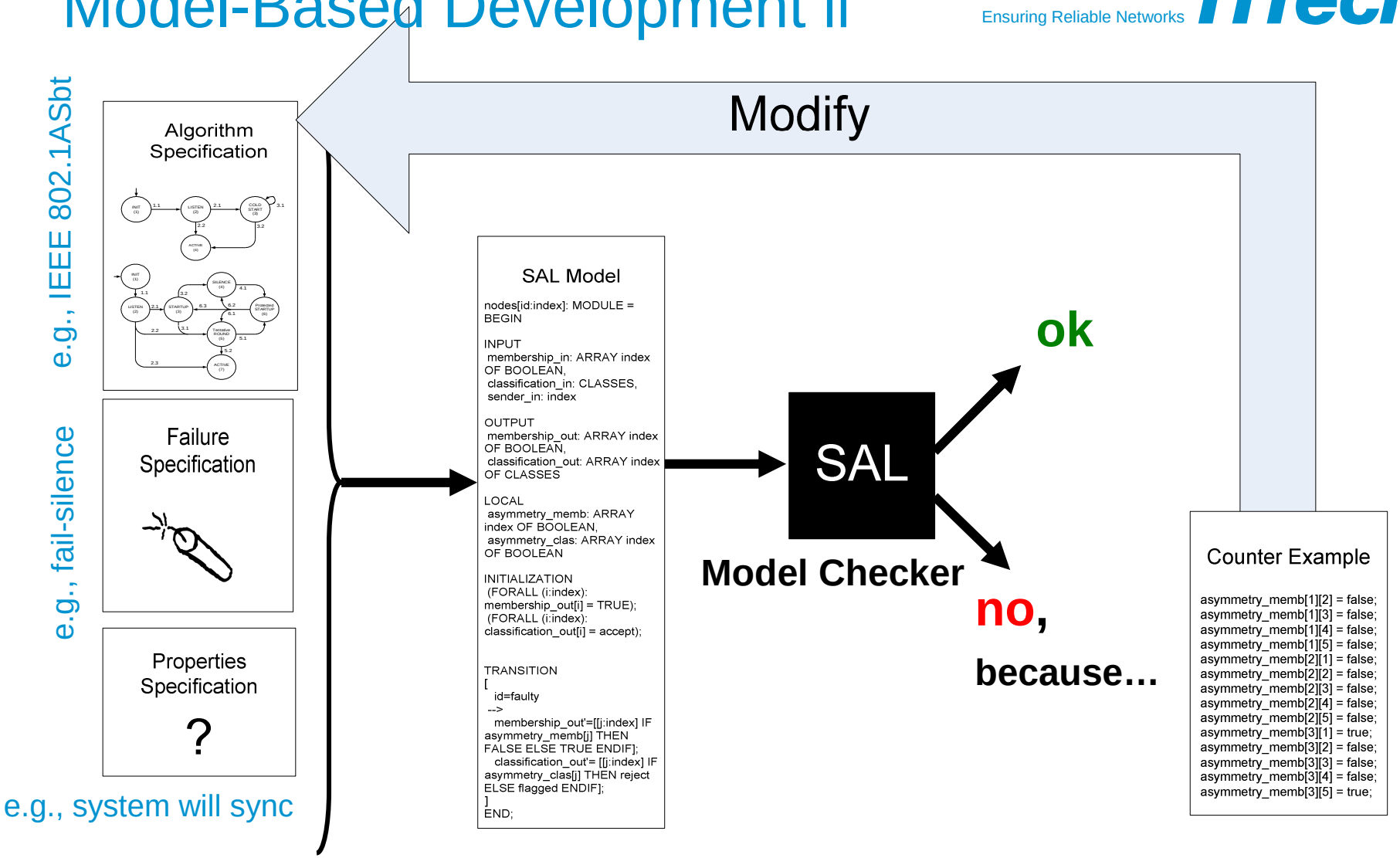
Development of fault-tolerant clock synchronization algorithms is non-trivial:

- synchronization proof is hard for certain failure modes
- completeness has to be proven as well
  - i.e., we need to prove that we have covered all possible failure scenarios

Therefore, formal methods are used in the development and in the verification of such algorithms.

- Theorem Proving is the process of developing a deductive proof, typically interactive with a proof assistant.
- Model Checking is an automatized approach.

Model-Based Development ii



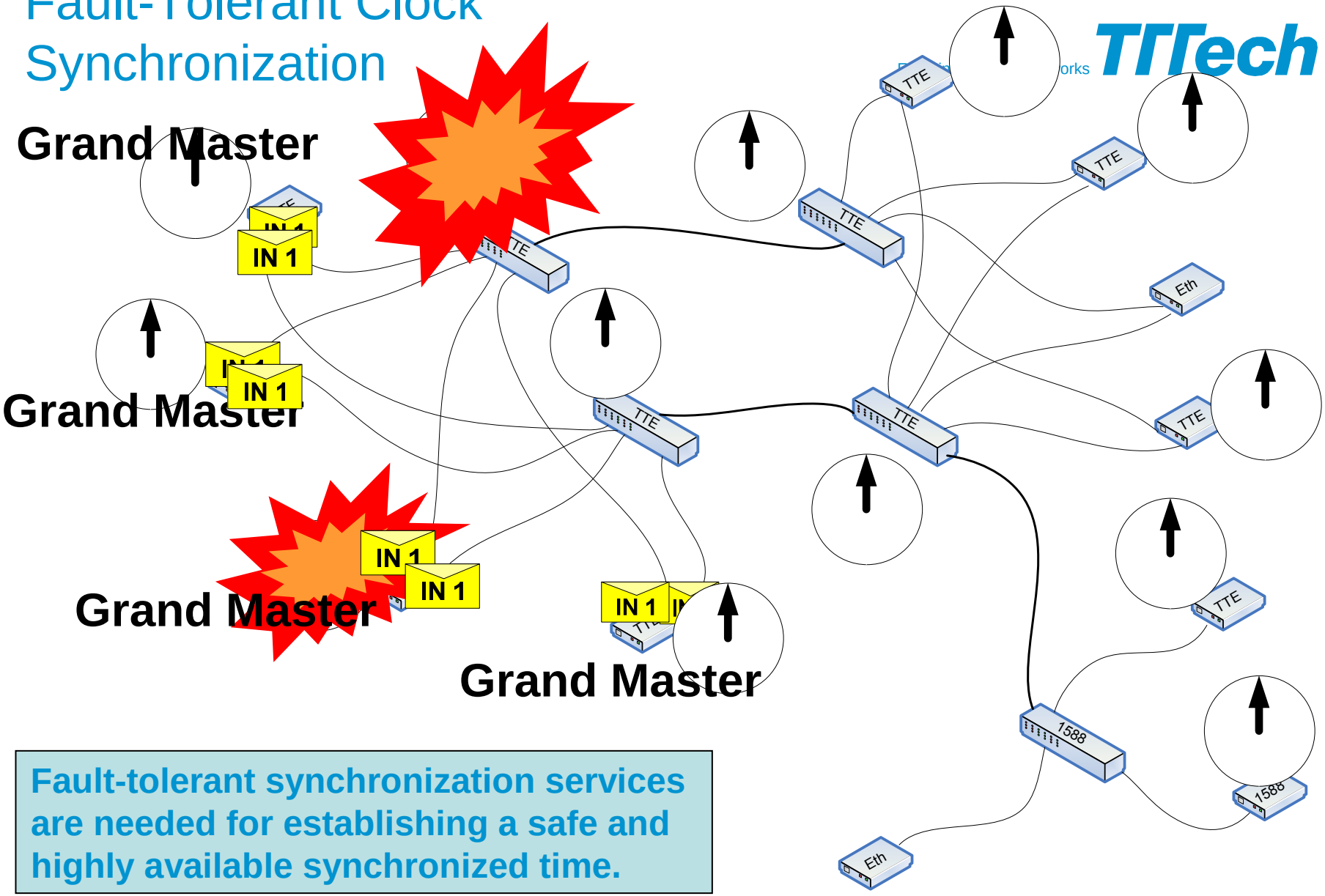
Example: SAE AS6802

First Byzantine fault-tolerant clock synchronization algorithm verified by model-checking only.

Basic algorithm addresses only synchronization of the clocks.

Extension for syntonization (we call it clock-rate correction) has been modeled and studied as well.

Fault-Tolerant Clock Synchronization



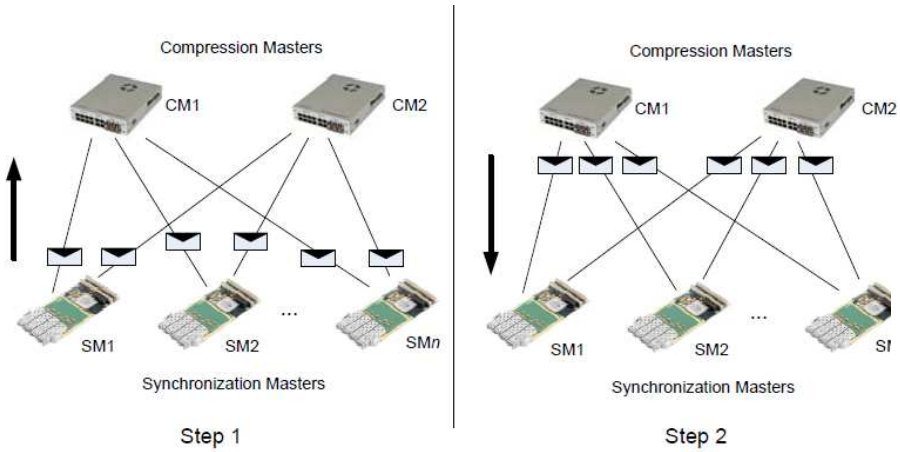
Fault-tolerant synchronization services are needed for establishing a safe and highly available synchronized time.

SAE AS6802 Clock Synchronization Algorithm

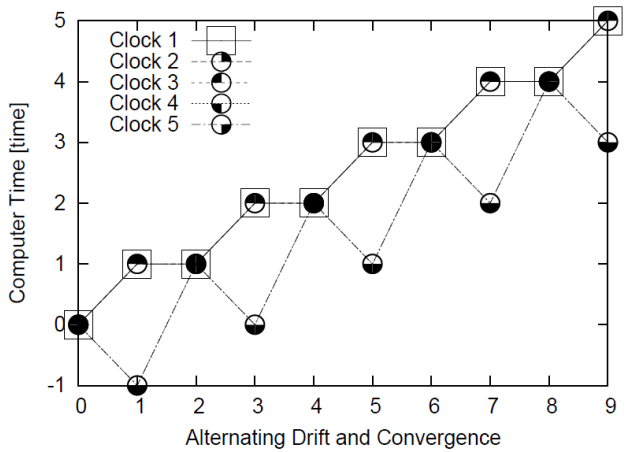
(case of five SM is updated in the standard)

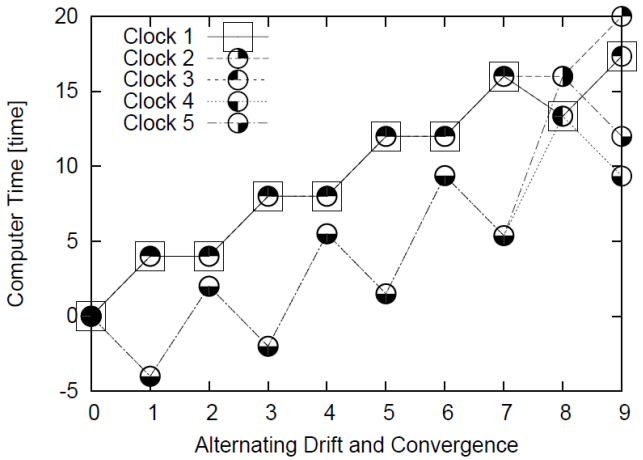
Algorithm Specification

- one SM clock: $compressed\_clock = SM\_clock_1$
- two SM clocks: $compressed\_clock = \frac{SM\_clock_1 + SM\_clock_2}{2}$
- three SM clocks: $compressed\_clock = SM\_clock_2$
- four SM clocks: $compressed\_clock = \frac{SM\_clock_2 + SM\_clock_3}{2}$
- five SM clocks: $compressed\_clock = SM\_clock_3$
- more than five SM clocks: take the average of the $(k+1)^{th}$ largest and $(k+1)^{th}$ smallest clocks, where k is the number of faulty SMs that have to be tolerated.



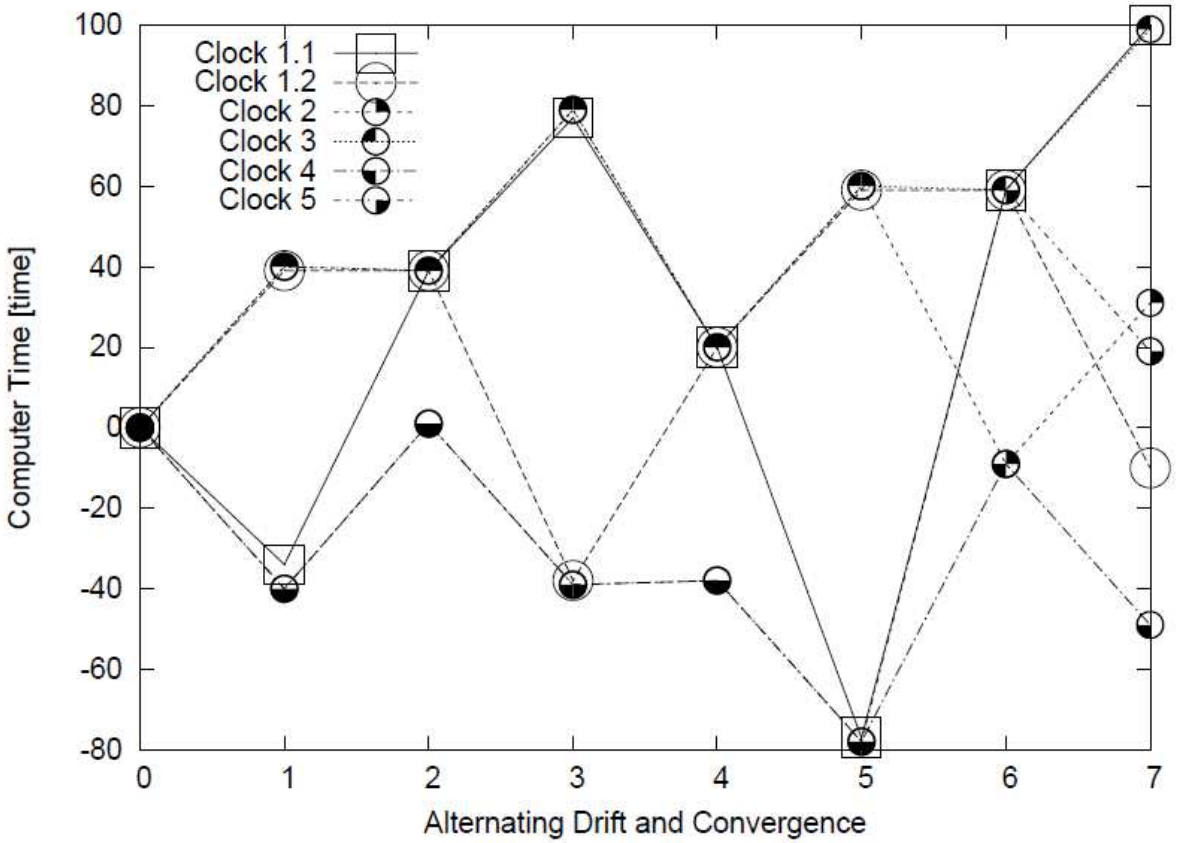
- one CM clock: $SM\_clock = CM\_clock_1$
- two CM clocks: $SM\_clock = \frac{CM\_clock_1 + CM\_clock_2}{2}$
- three CM clocks: $SM\_clock = CM\_clock_2$



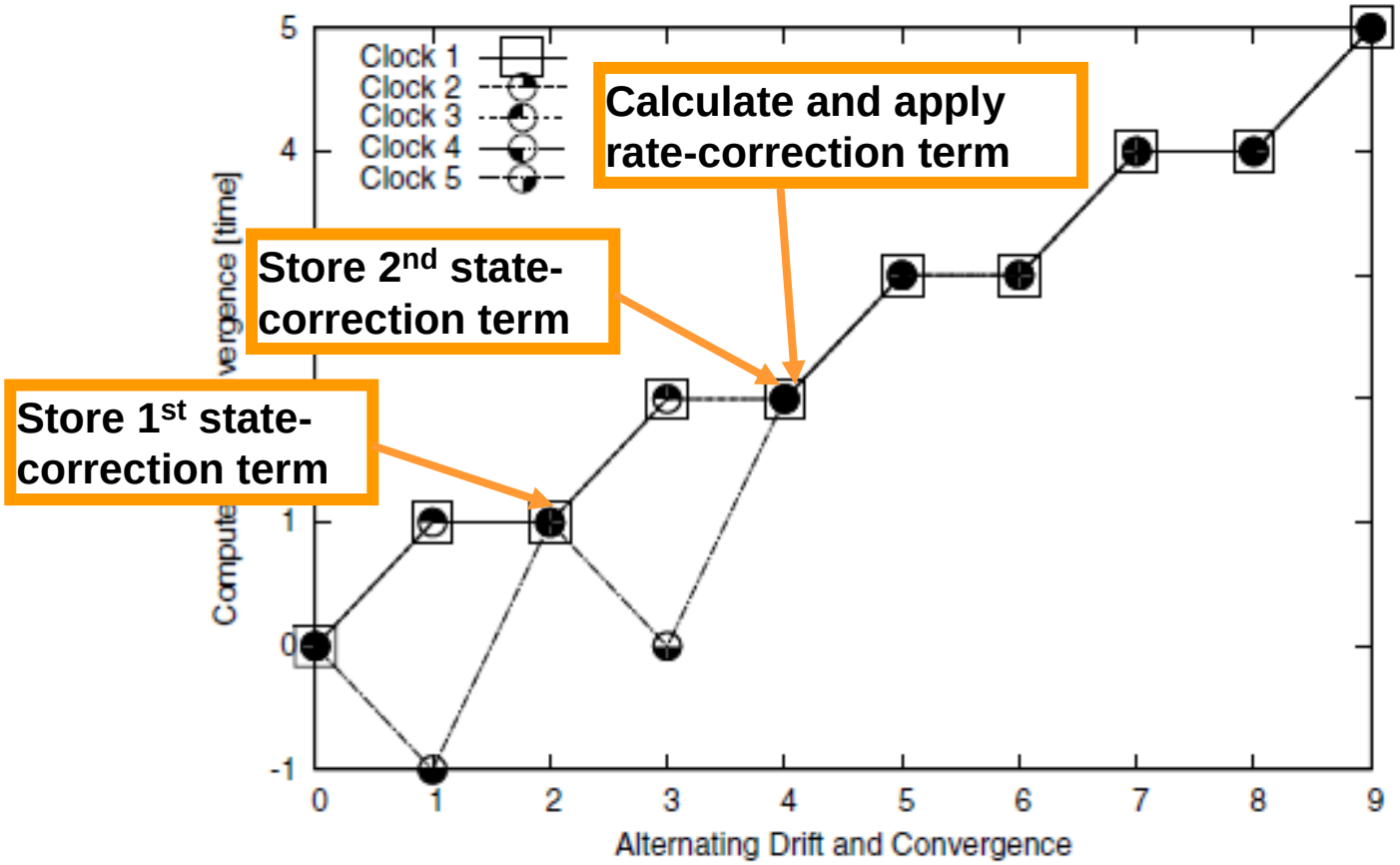


Byzantine Failure Tolerance

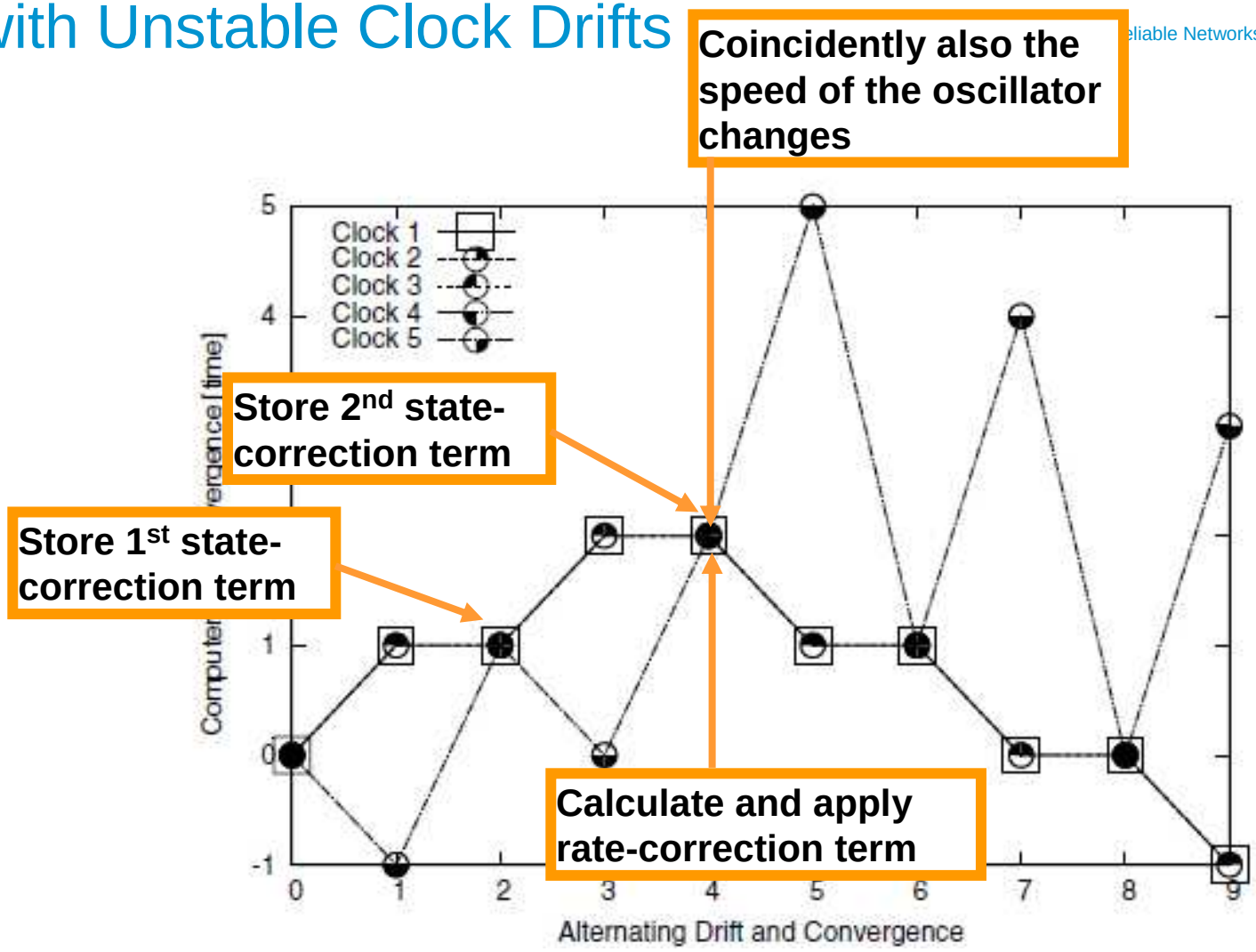
Occurrence of a Byzantine failure is a combination of a fail-arbitrary synchronization master (end station) and an inconsistent-omission faulty compression master (bridge).



Rate-Correction with Stable Clock Drifts



Rate-Correction with Unstable Clock Drifts



What are the failure modes of IEEE 802.1ASbt

Permanent fail-silence?

Transient/Intermittent fail-silence?

Fail-consistent faulty?

- e.g., a grandmaster providing faulty time

Inconsistent faulty bridges?

- e.g., a bridge forwarding time information only on some ports

Byzantine faulty grandmaster clocks?

TTTech

Ensuring Reliable Networks

www.tttech.com

Wilfried Steiner, Corporate Scientist
wilfried.steiner@tttech.com

Backup

TTTech

Ensuring Reliable Networks

www.tttech.com

Static vs. Dynamic Systems

Situation:

What is the color of the house?

Static Situation – one Truth

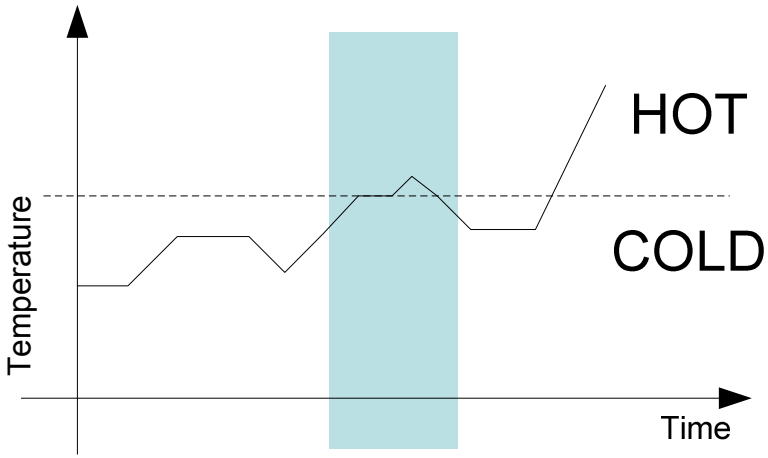
Situation:

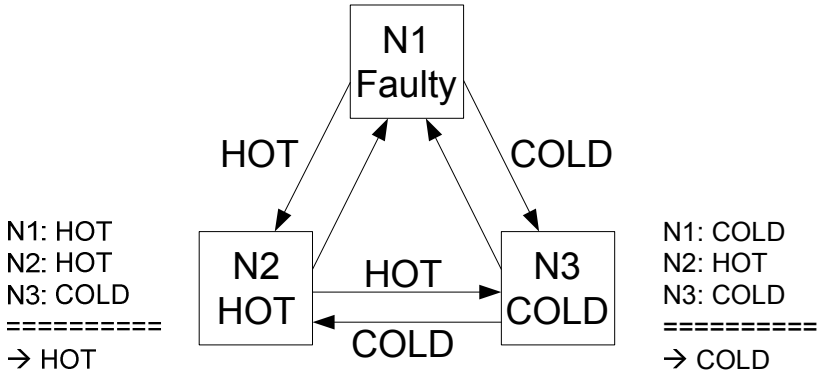
What is the color of the ball ?

Dynamic Situation – >one Truth

Origins: Byzantine Failures

A distributed system that measures the temperature of a vessel shall raise an alarm when the temperature exceeds a certain threshold.
The system shall tolerate the arbitrary failure of one node.
How many nodes are required?
How many messages are required?

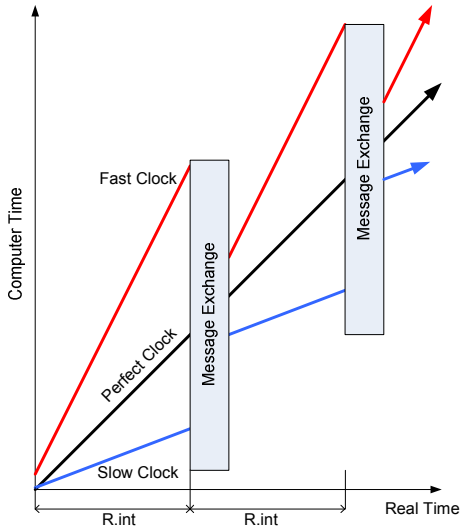


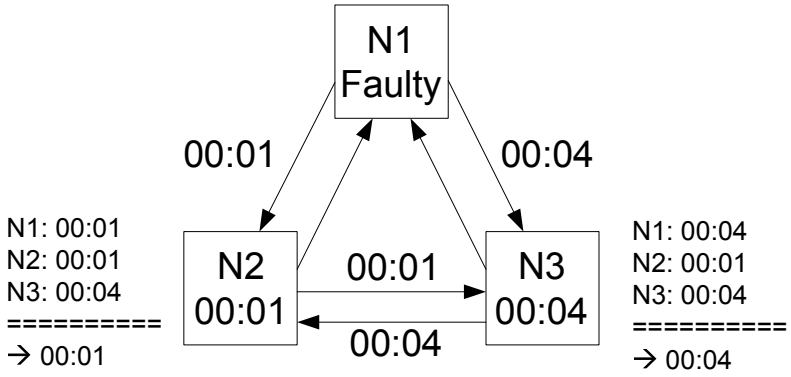


In general, three nodes are insufficient to tolerate the arbitrary failure of a single node.
The two correct nodes are not always able to agree on a value.
A decent body of scientific literature exists that address this problem of dependable systems, in particular dependable communication.

Byzantine Clocks

A distributed system in which all nodes are equipped with local clocks, all clocks shall become and remain synchronized.
The system shall tolerate the arbitrary failure of one node.
How many nodes are required?
How many messages are required?





In general, three nodes are insufficient to tolerate the arbitrary failure of a single node.
The two correct nodes are not always able to bring their clocks into close agreement.
A decent body of scientific literature exists that address this problem of fault-tolerant clock synchronization.