TU Wien

Systems of Systems Need a Global Timebase

H.Kopetz

June 2014

Work supported by EU FP 7 Project AMADEOS grant agreement 610535

Outline

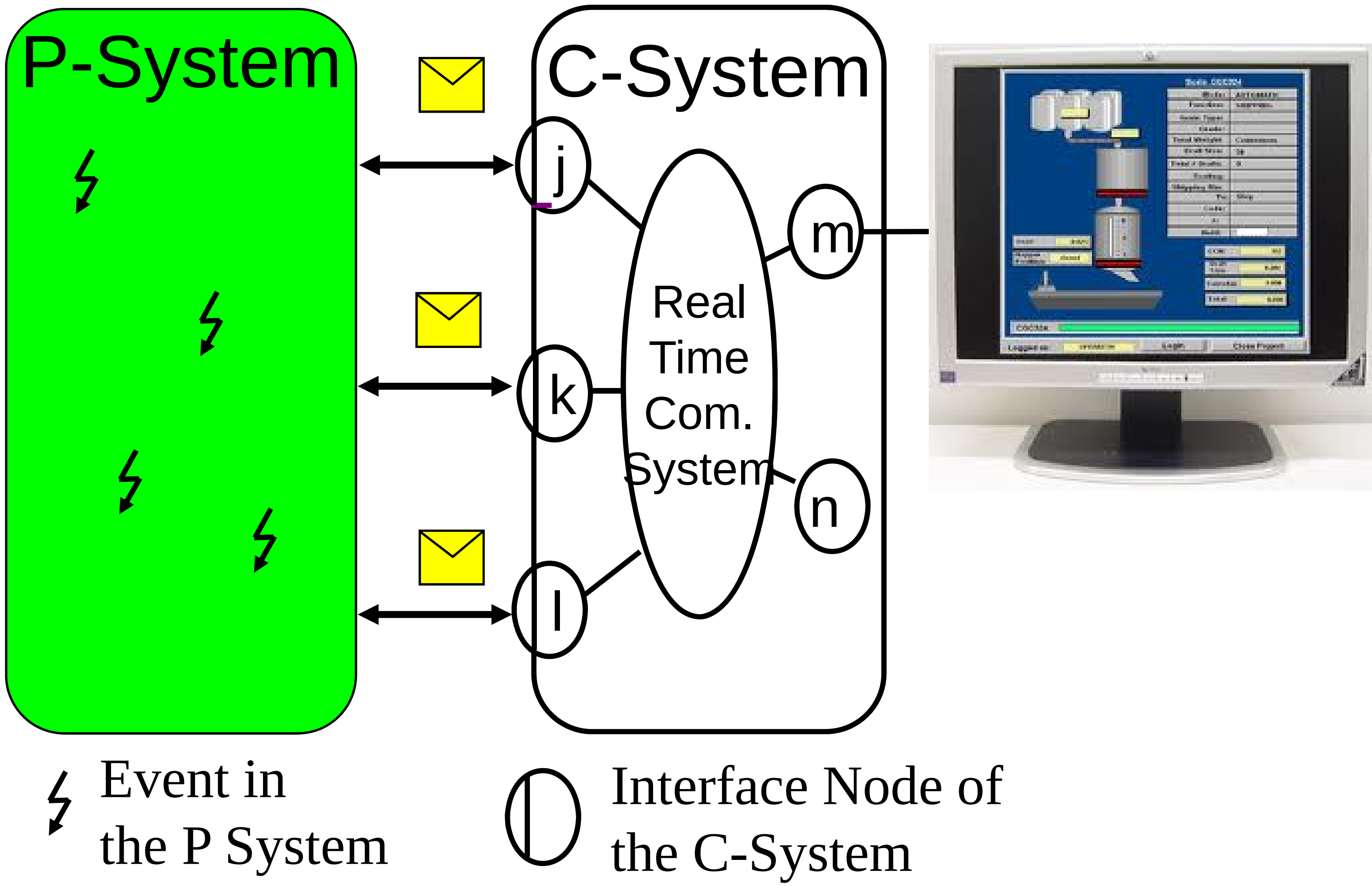
- Introduction
- Why is an SoS *Different*?
- Why is a Global Time *Needed* in an SoS?
- What *Properties* of the Global Time Base are Required in an SoS?
  - Adequate Precision
  - Sparseness
  - Synchronized with TAI
  - Long Holdover
  - Fault-Tolerance
- How to *Distribute* the Global Time in an SoS?
- Conclusion

System of Systems (SoS)

An SoS is an integration of a finite number *of autonomous constituent systems (CS) e.g., embedded systems,* which are independent and operable, and which are networked together for a period of time to achieve a certain higher goal (refer to Jamshidi, 2009, T-Area SoS)**.**

An autonomous constituent system (CS) encompasses a computer system, a physical object and a human operator—it is a cyber-physical system with *non-deterministic* behavior .

CPS: *Physical World* meets *Cyber World*



Physical (P) System versus *Cyber (C) System*

P-System

Controlled by the laws of physics

Physical time

Time base dense

C-System

Controlled by program execution

Execution time

Time-base discrete

The model of the P-system that is used by the C-system must be aware of the progression of Physical time.

Characteristics of a System of Systems (SoS)

Characteristic	***Old* System**	**SoS**
Scope of System	Fixed (known)	Not known
Clock synchronization	Internal	External (GPS)
Structure	Hierarchical	Networked
Requirements and Spec.	Fixed	Changing
Evolution	Version control	Uncoordinated
Testing	Test phases	Continuous
Implementation Technology	Given and fixed	Unknown
Faults (Physical, Design)	Exceptional	Normal
Control	Central	Autonomous
Emergence	Insignificant	Important
System development	Process model	???

Key Issues in SoS Design

- Global Time
- Information Representation
- Evolution
- Emergent Phenomena
- Faults are *Normal* (Security and Safety)
- *Cognitive* Complexity

Data vs. Information

NASA's Mars Climate Orbiter was lost in space . . . because engineers failed to make a simple conversion from English units to metric, an embarrassing lapse that sent the $125 million craft fatally close to the Martian surface, investigators said yesterday.

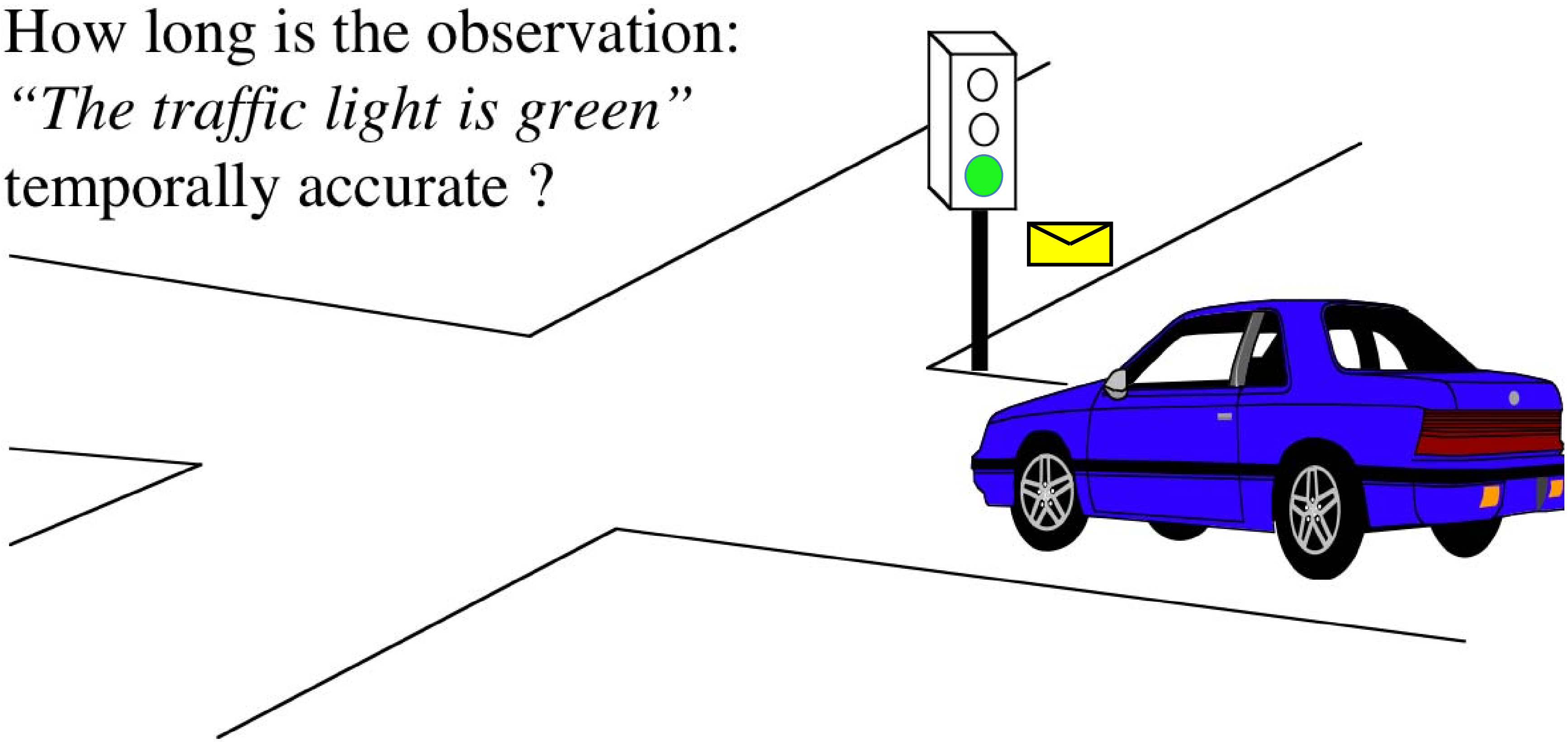
By Kathy Sawyer, Washington Post Staff Writer, Friday, October 1, 1999; Page A1

The same *data* can convey a differing *information* in differing contexts (Different CSs of an SoS).

Global Time is Needed in an SoS to

- Enable the interpretation of timestamps in the different CSs of the SoS
- Limit the validity of real-time control data
- Specify the temporal properties of Interfaces
- Synchronize Input and Output Actions across the SoS
- Provide conflict-free Resource Allocation
- Perform prompt Error Detection
- Temporal Error Containment
- Strengthen Security Protocols

Car 2 Infrastructure:



An appropriate model of RT communication must consider *timeliness* as important as *correctness*.

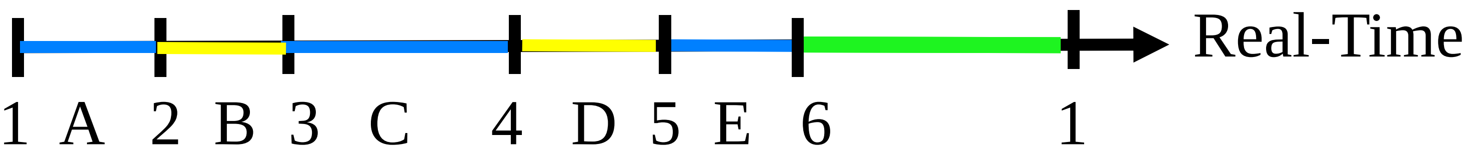
On *Timestamps*

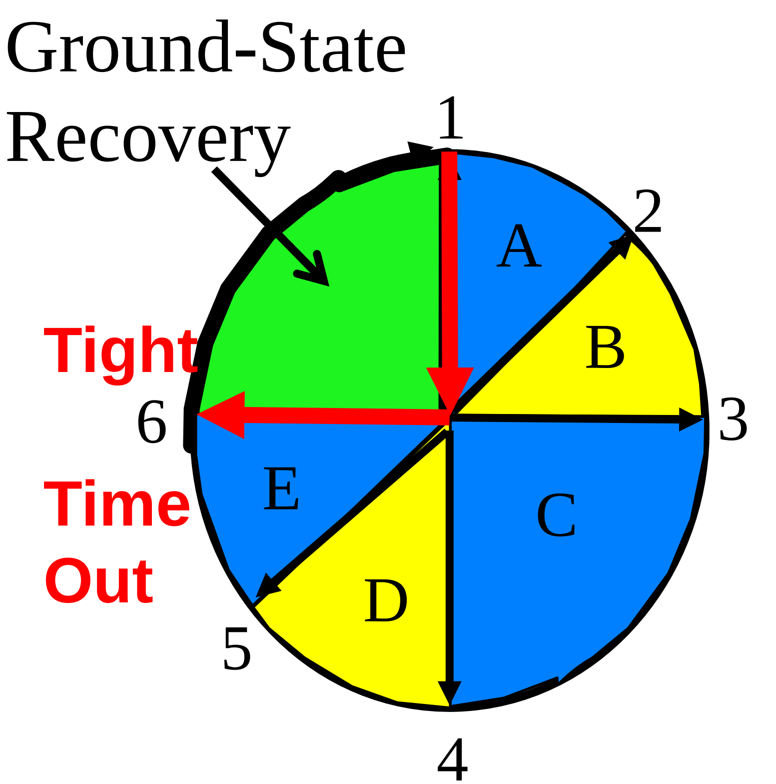
- A timestamp records (on *input*) or determines (on *output*) the occurrence of a *physical event* by observing the state of a physical clock at the instant of event occurrence.
- Timestamps must be taken at the interface between the physical world, the site of *event occurrence*, and cyberspace.
- Any delay (or, worse, *jitter*) in taking a timestamp leads to a loss of temporal precision. Many wireless protocols (e.g., WIFI) have a significant jitter.
- A timestamp taken by one clock is only meaningful to a subsystem with a different clock if the clocks are synchronized.

Example: August 14, 2003 Blackout Report

A valuable lesson from the August 14 blackout is **the importance of having *time-synchronized system data recorders***. *The Task Force's investigators labored over thousands of data items to determine the sequence of events, much like putting together small pieces of a very large puzzle. That process would have been significantly faster and easier if there had been wider use of synchronized data recording devices*. From Final Report on US-Canadian August 14, 2003 Power Blackout, p.164.

Synchronize Actions in an SoS

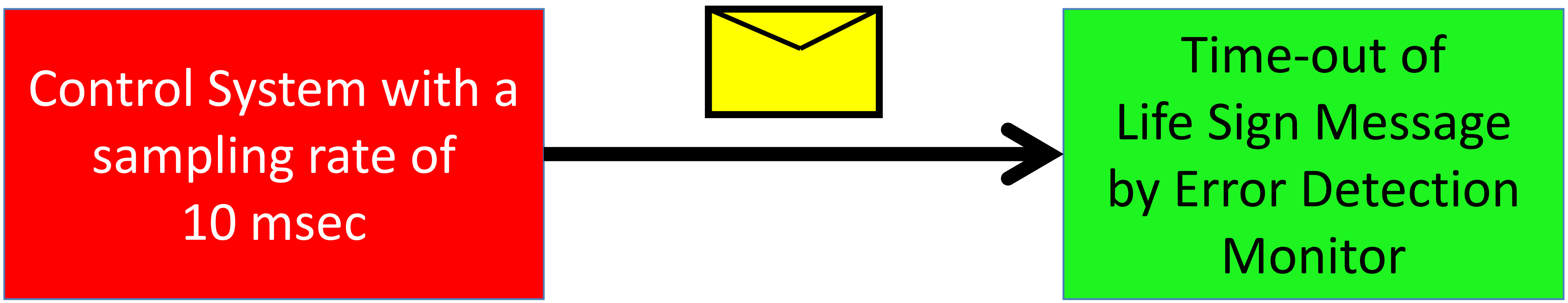




1 Start of Cycle
A Observation of Sensor Input
2 Start of Transmission of Sensor Data
B Transmission of Input Data
3 Start of Processing of Control Algorithm
C Processing of Control Algorithm
4 Termination of Processing
D Transmission of Output Data
5 Start of Output to Actuators
E Output Operation at the Actuator
6 Termination of Output Operation

Prompt Error Detection

Periodic Life Sign Message every cycle



Maximum Error Detection Latency of Fail-Silent Failure:

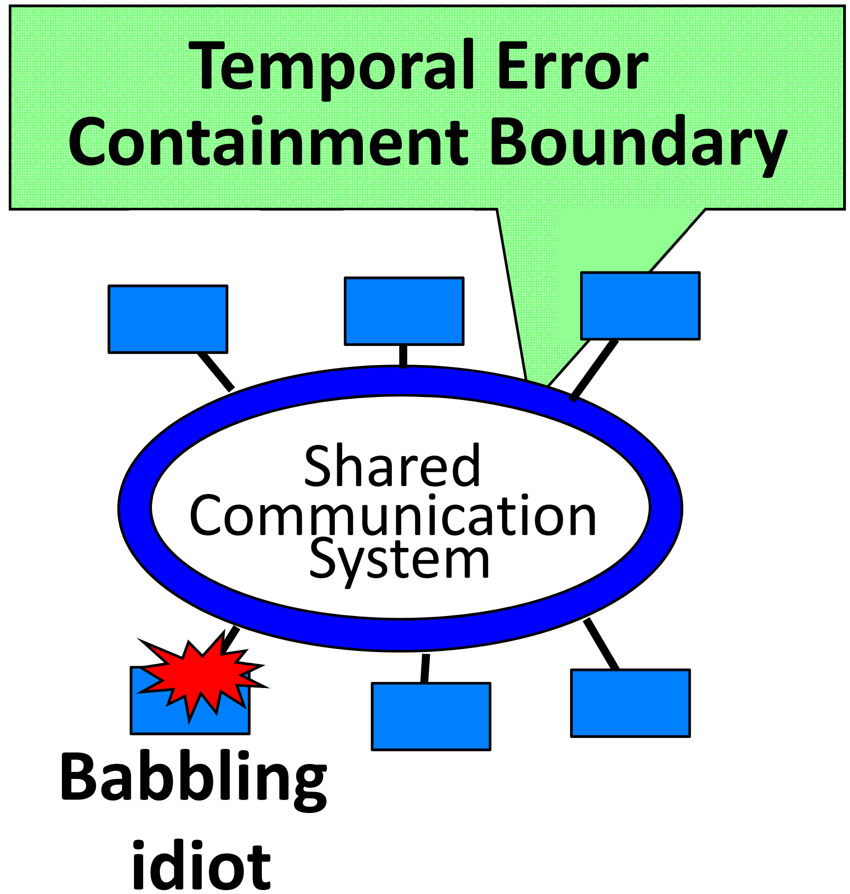
Prompt Error Detection reduces recovery time.

Temporal Error Containment

It is ***impossible*** to maintain the communication among the correct components of a RT-cluster **if the temporal errors caused by a faulty component are not contained**.

Error containment of an *arbitrary temporal node failure requires that the shared* Comm. System is a self-contained FCU that has *temporal information* about the allowed behavior of the nodes–

It must contain application-specific state.



High priority message in a CAN System?

Many SoS Applications Rely on GPS Time

Many of the existing SoS Infrastructure Applications globally synchronize their clocks by GPS time.

Examples:

- Airlines use GPS for Navigation (instead of ILS)
- Utility Companies use GPS to synchronize the clocks of the PMUs (Phase Measurement Units).
- Telecommunication Companies use GPS to synchronize the clocks in the base stations.

In a recent US GAO report (GAO 14-15) on *GPS Disruptions* (Nov. 2013) the reliance of a significant part of the US infrastructure on the *GPS* is discussed.

Properties of the Global Time in an SoS

- Adequate Precision
- Sparseness
- Synchronized with TAI
- Long Holdover
- Fault-Tolerance

Precision versus *Accuracy* of a Time-Base

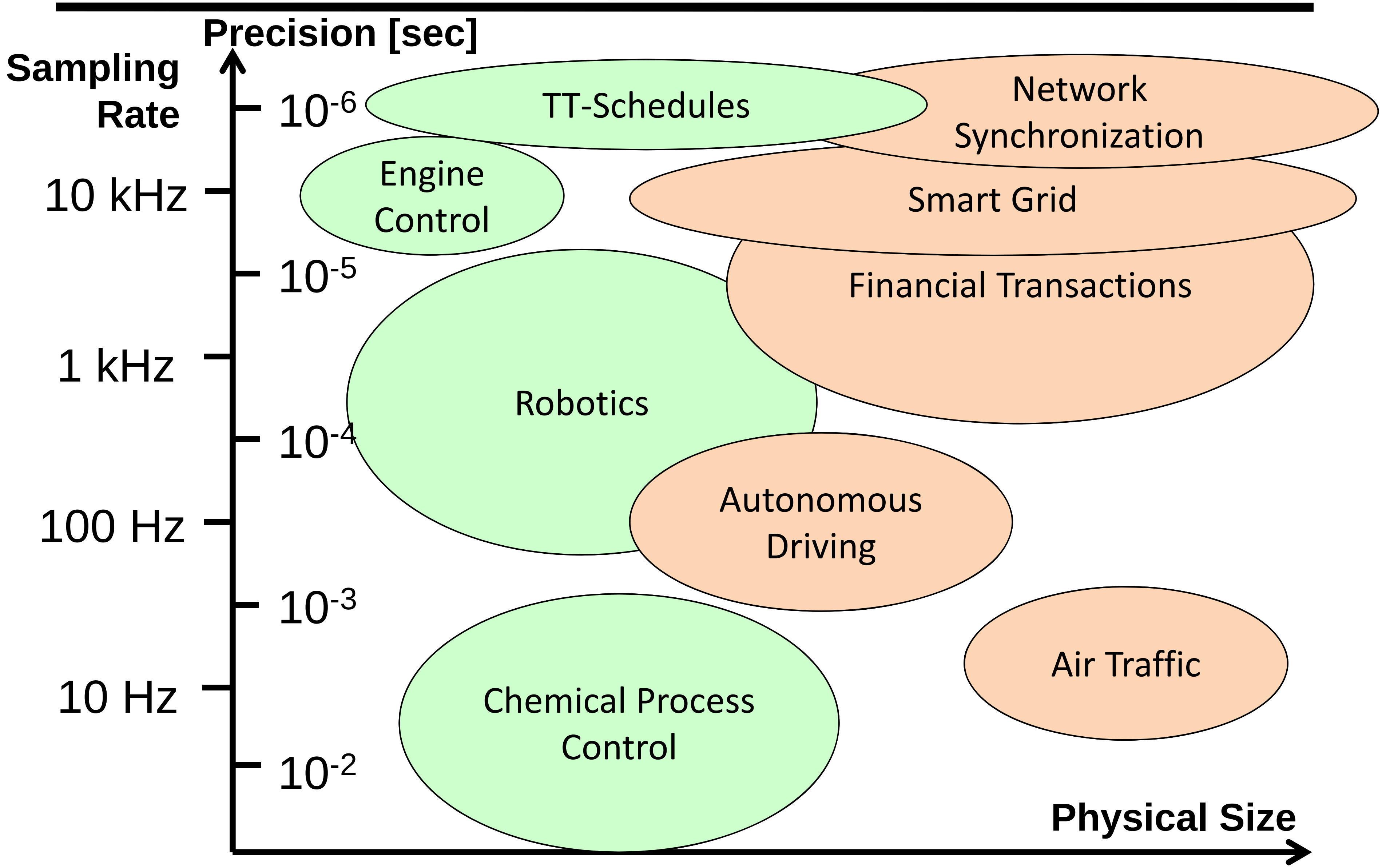
Let us assume, there exists an *omniscient reference clock in a distributed control system*:

Precision: The maximum difference between any two respective *good* ticks of an ***ensemble of clocks*** during the *Interval of Discourse*, measured by the reference clock.—***Internal clock synchronization***

Accuracy: The maximum difference between any *good* tick of clock and the respective tick of ***a common reference clock*** during the *Interval of Discourse*, measured by the reference clock.—***External clock synchronization****.*

Precision ≤ 2 Accuracy

Precision Requirements



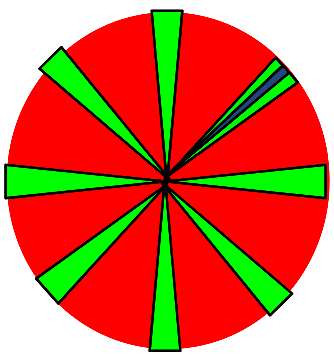
Models of Time in the CPS

Dense

Physics

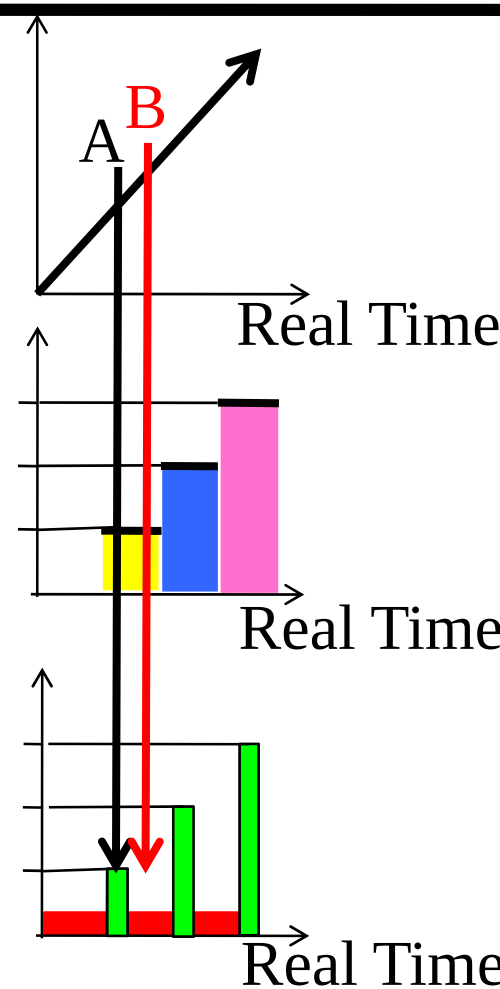
Discrete

Central Computer



Sparse

Distributed Computer



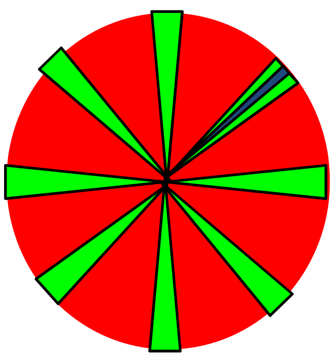
Models of Time in the CPS

Dense

Physics

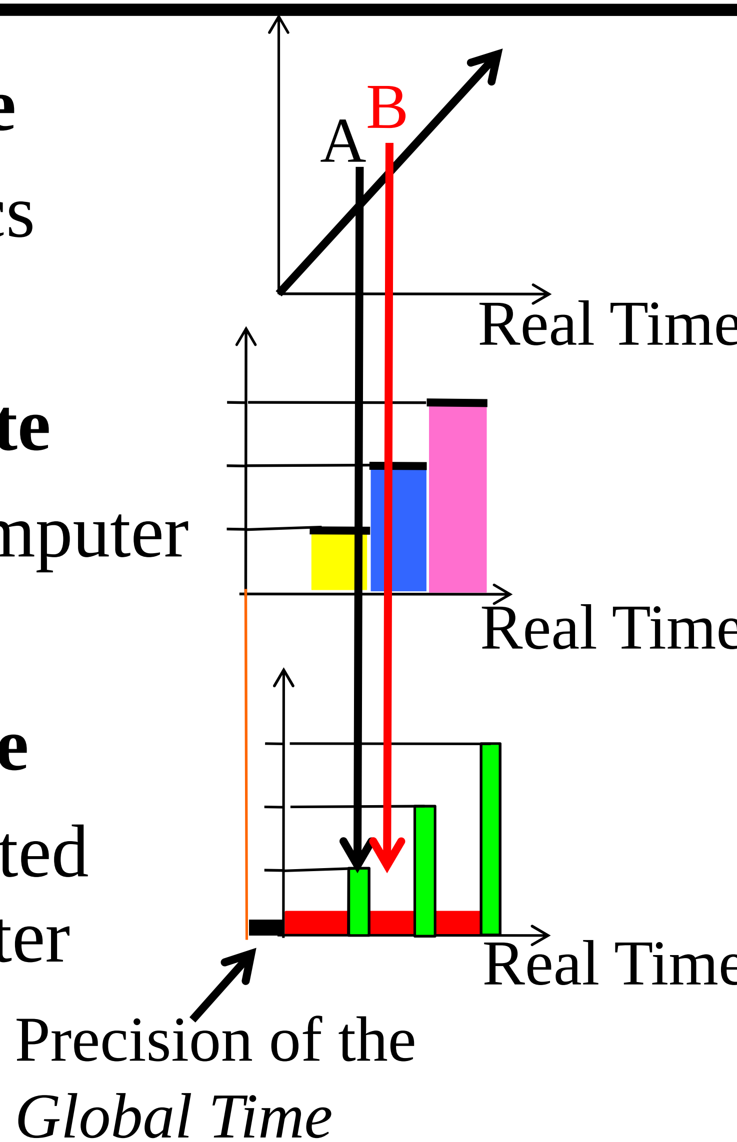
Discrete

Central Computer



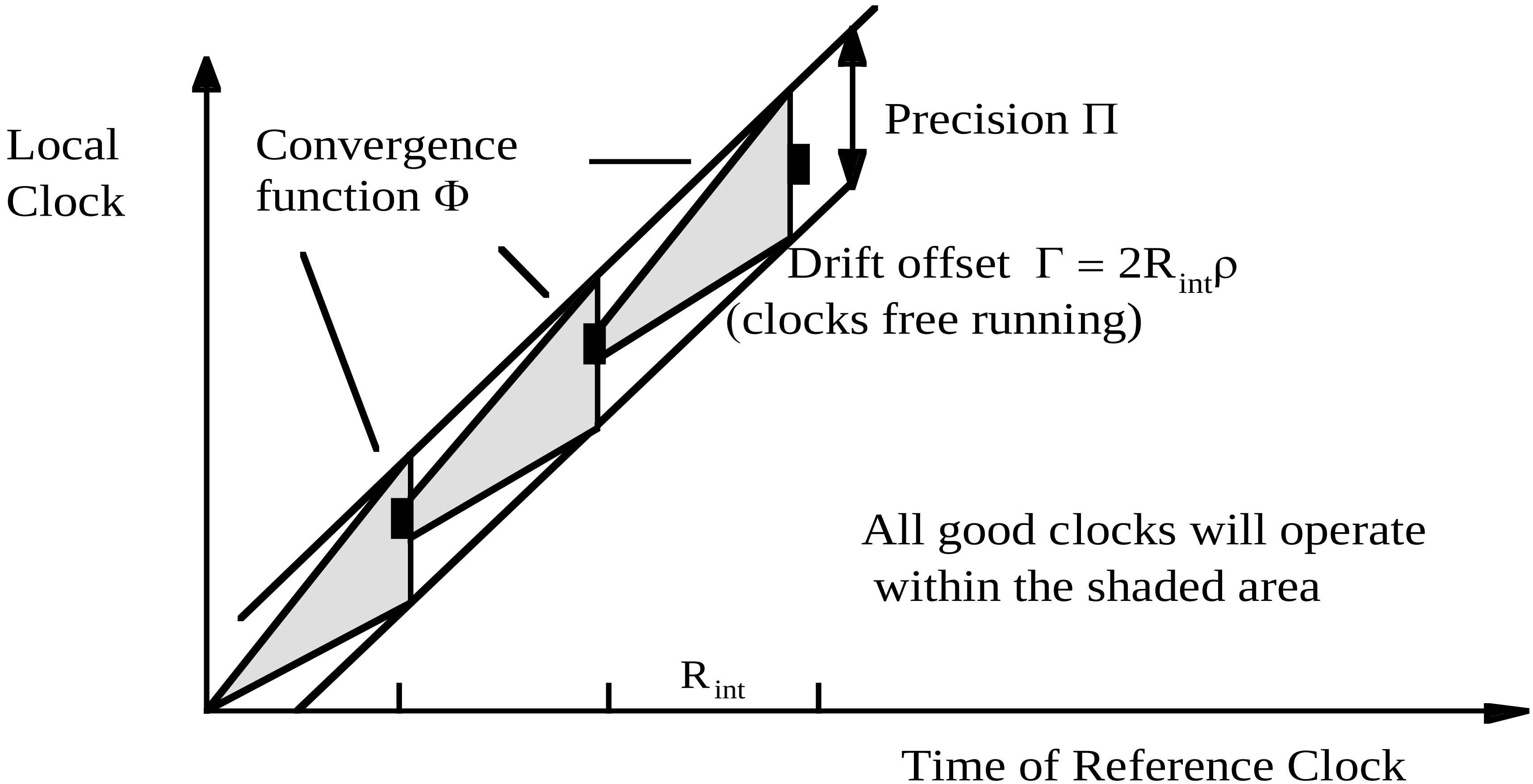
Sparse

Distributed Computer

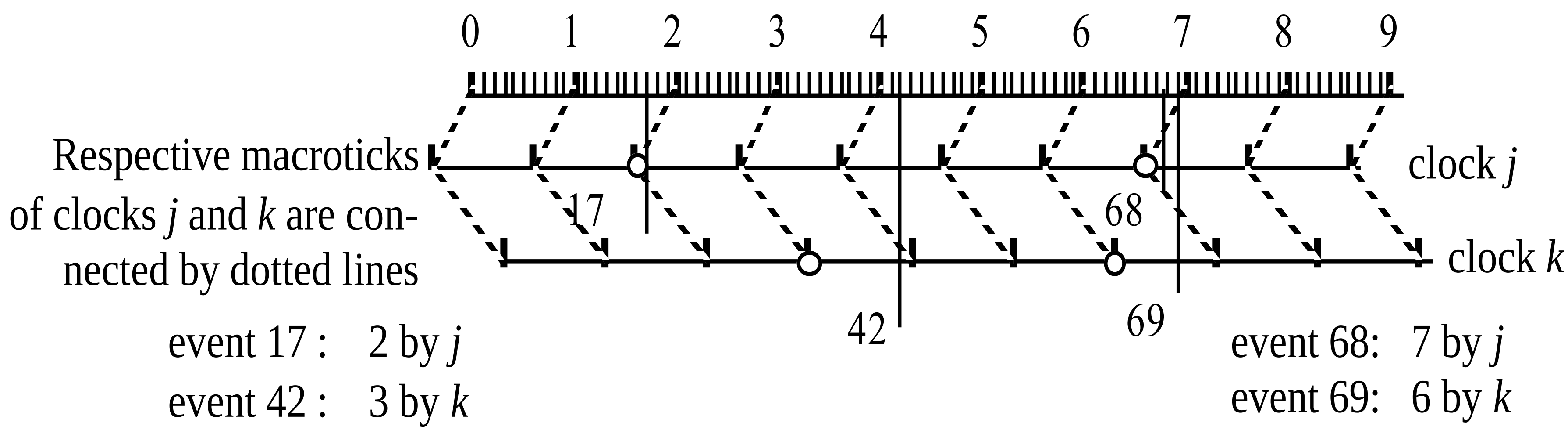


Clock Synchronization Condition

$$\Phi + \Gamma \leq \Pi$$

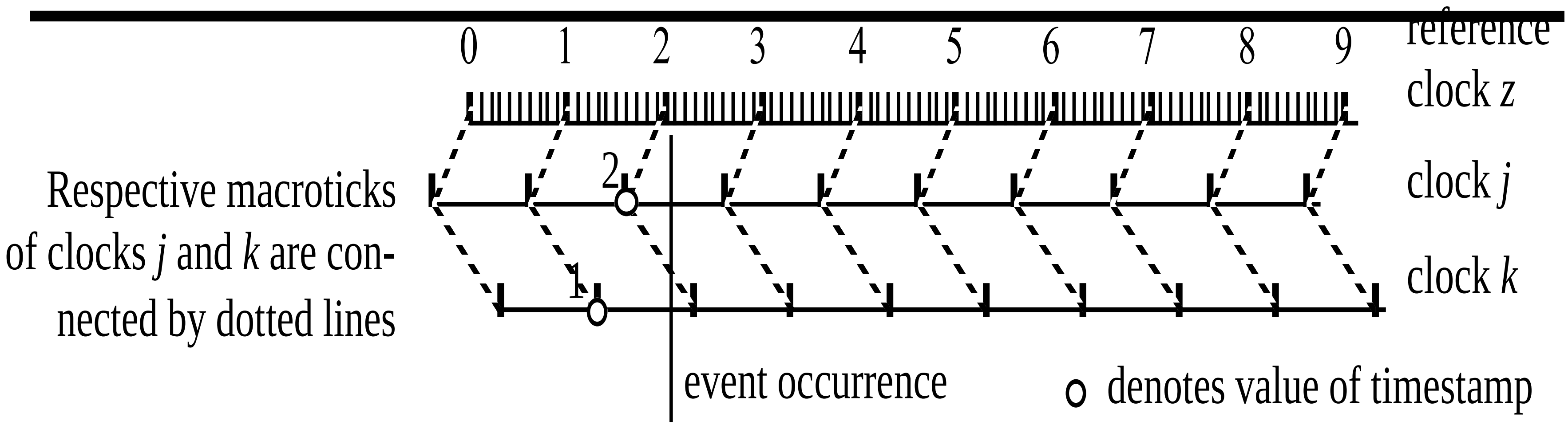


One Tick Difference: What Does it Mean?



Because of the accumulation of the synchronization error and the digitalization error, it is not possible to reconstruct the temporal order of two events from the knowledge that the global timestamps differ by one tick.

Reasonableness Condition



The global time t is called *reasonable*, if all local implementations of the global time satisfy the following reasonableness condition for the global granularity g of a macrotick:

$$g > \Pi$$

This reasonableness condition ensures that the synchronization error is bounded to less than one macrogranule, i.e., the duration between two macroticks.

Fundamental Limits to Time Measurement

Given a distributed system with a reasonable global time base with granularity g. Then the following fundamental limits to time measurement must be observed:

- If a single event is observed by two nodes, there is always the possibility that the timestamps will differ by one tick
- Let us assume that d_{obs} is the observed duration of an interval. Then the true duration d_{true} is
  $$(d_{obs} - 2g) < d_{true} <(d_{obs} + 2g)$$
- The temporal order of events can only be recovered, if the observed time difference $d_{obs} \geq 2g$
- The temporal order of events can always be recovered, if the event set is 0/3g precedent.

An *Impossibility Results*

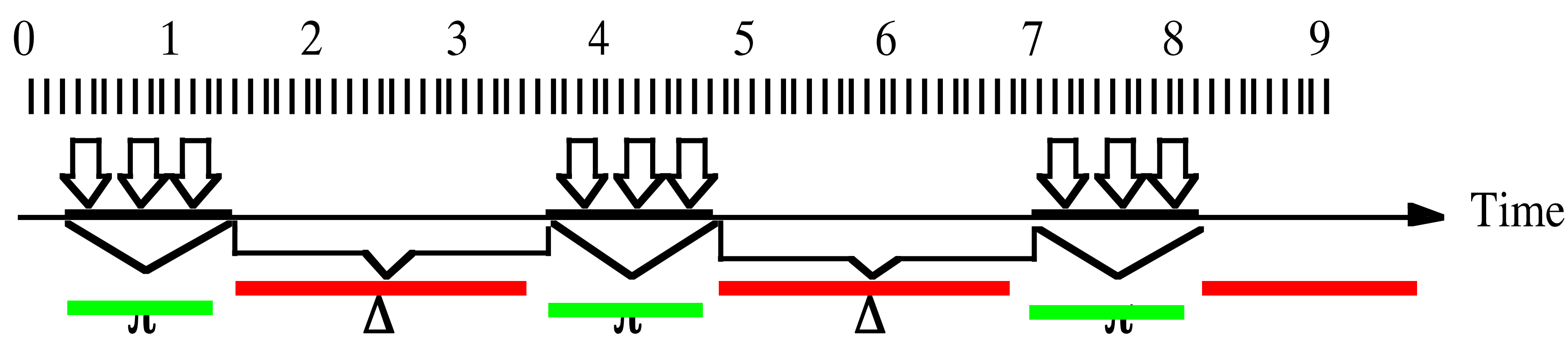
It is impossible to represent the temporal properties of the dynamic *analog* physical world with *true fidelity* in *digital* cyberspace.

- The conflict between *fidelity* and *consistency* can be reduced, but can never be fully resolved.
- The better the precision of the clock synchronization, the smaller the error introduced by digitalization and synchronization.

The problem: *consistency* versus *fidelity*

Sparse Time Model

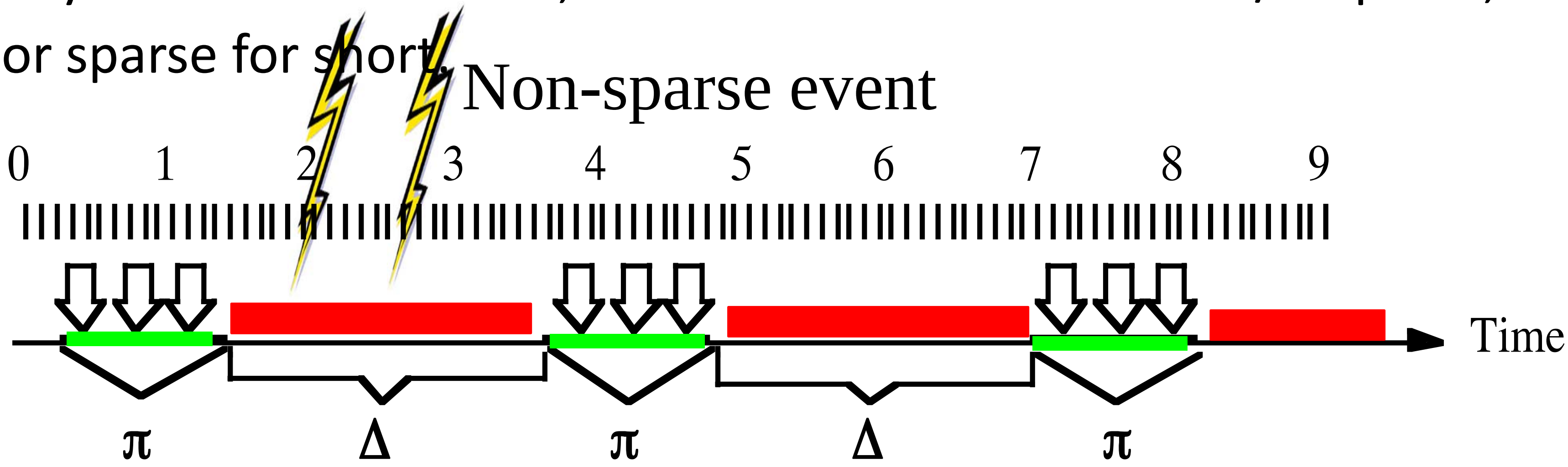
If the occurrence of events is restricted to some active intervals on the timeline with duration π with an interval of silence of duration Δ between any two active intervals, then we call the time base π/Δ-sparse, or ***sparse*** for short, and events that occur during the active intervals ***sparse events*** and , and events that occur outside the active intervals ***non-sparse events.***



Events ⇩ are only allowed to occur at subintervals of the timeline

Events *outside* the SoC: Agreement Protocols

If the occurrence of events is restricted to some active intervals with duration π with an interval of silence of duration Δ between any two active intervals, then we call the time base π/Δ-sparse, or sparse for short.



Events ⇩ are only allowed to occur at subintervals of the timeline

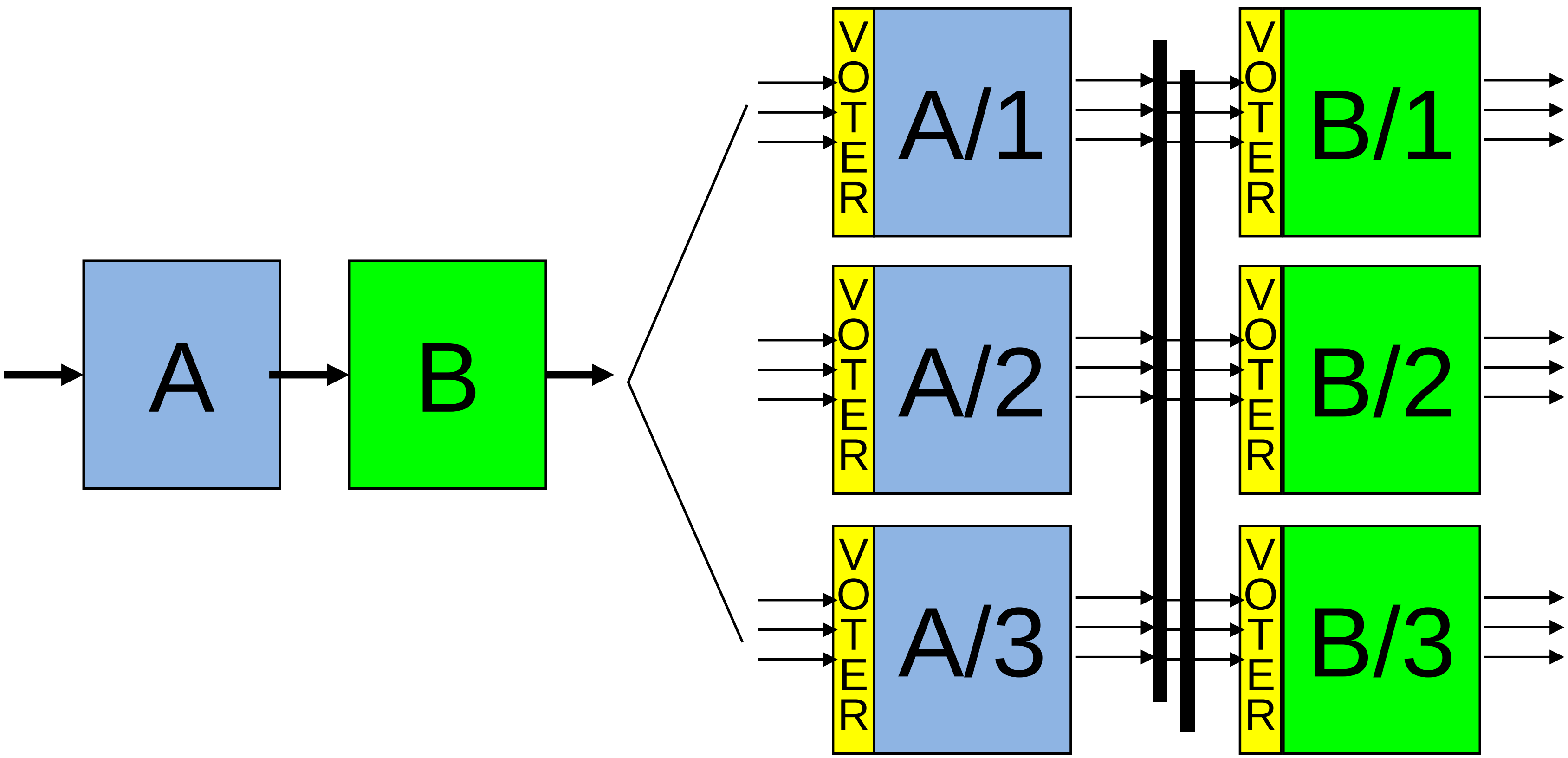
Agreement Protocol to Generate *Sparse Events*

To arrive at a consistent view of the temporal order of non-sparse events within a distributed computer system (which does not necessarily reflect the temporal order of event occurrence), the nodes must execute an *agreement protocol*.

(i) exchange information about the observations among all nodes, such that all nodes have the same data set.
(ii) every node executes the same algorithm on this data set to arrive at a consistent value and at a sparse interval of the observation.

Triple Modular Redundancy (TMR)

Triple Modular Redundancy (TMR) is the *generally accepted technique* for the mitigation of component failures at the system level:



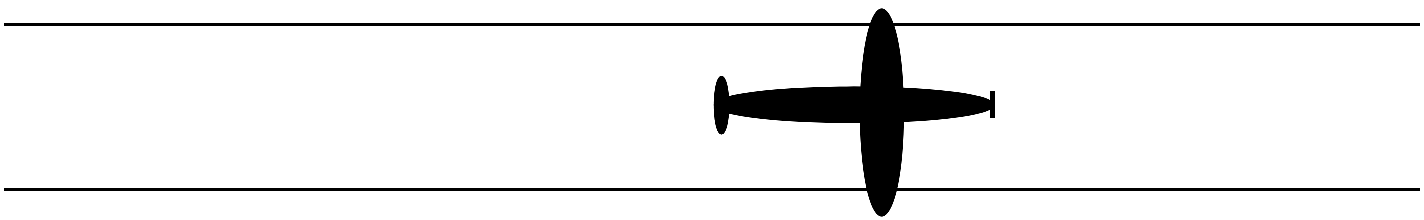
What is Needed to Implement TMR?

What architectural services are needed to implement Triple Modular Redundancy (TMR) at the architecture level?

- Provision of an Independent Fault-Containment Region for each one of the replicated components
- **Synchronization Infrastructure for the components**
- Predictable Multicast Communication
- Replicated Communication Channels
- Support for Voting
- **Replica Deterministic** (*which includes timely*) Operation
- Identical state in the distributed components

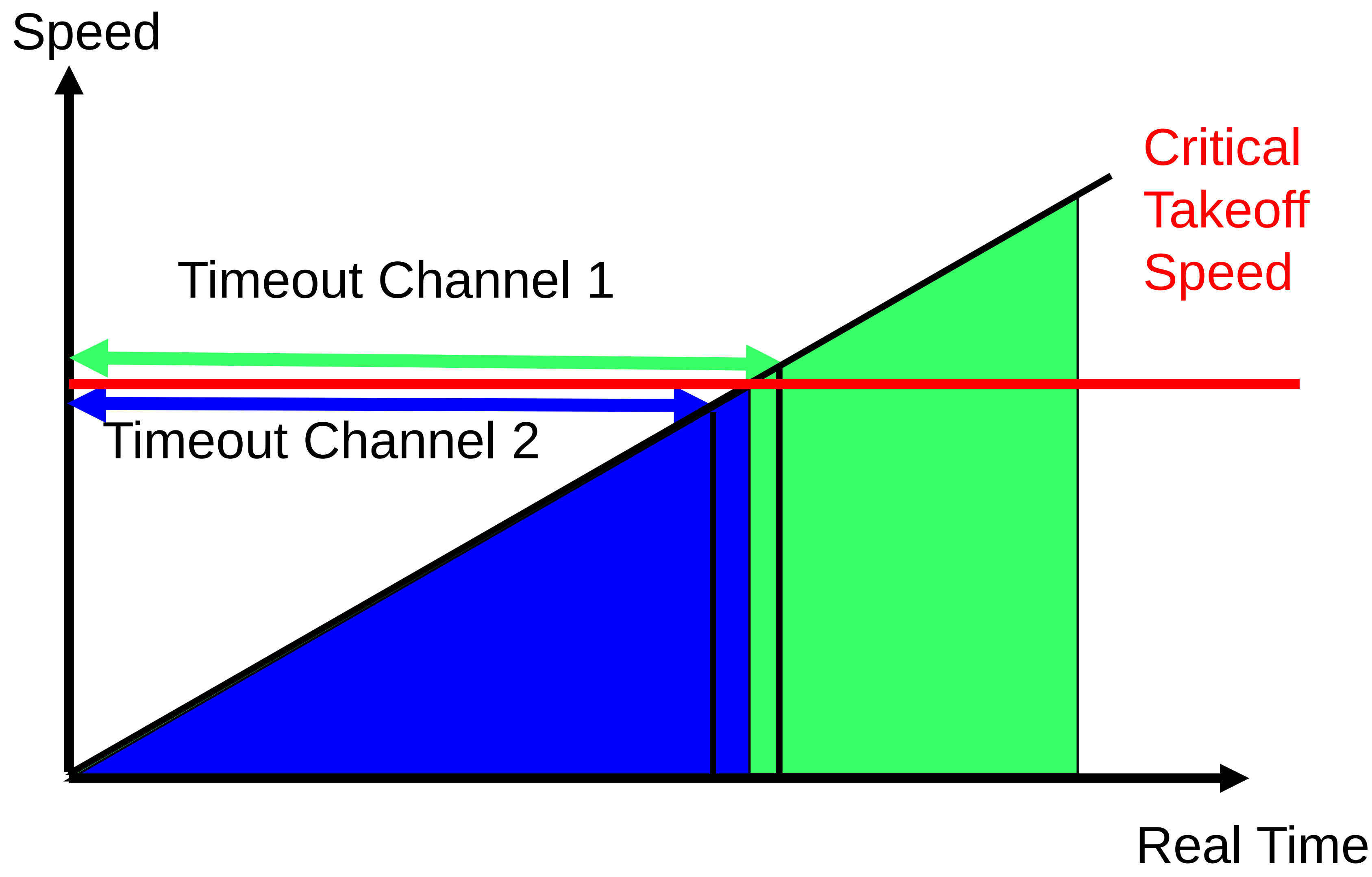
Replica Determinism: Airplane on Takeoff

Consider an airplane that is taking off from a runway with a flight control system consisting of ***three independent channels*** without a global time. Consider the system at the *critical instant* before takeoff:



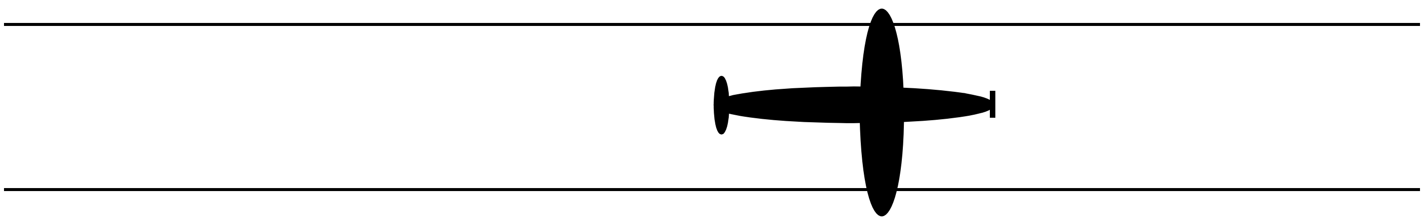
Channel 1	**Take off**	**Accelerate Engine**
Channel 2	**Abort**	**Stop Engine**

The Critical Role of Time



Replica Determinism: Airplane on Takeoff

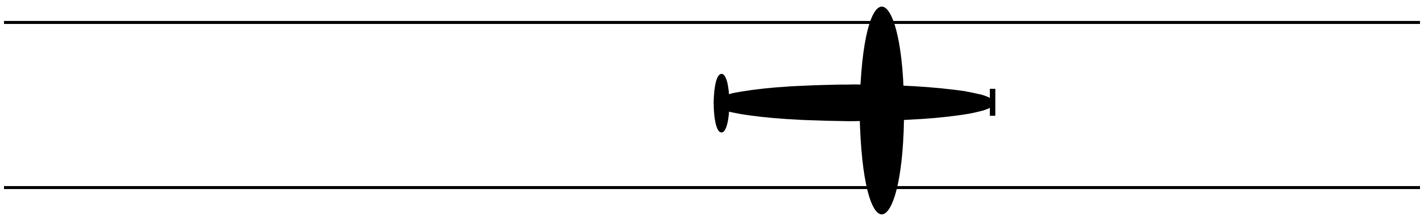
Consider an airplane that is taking off from a runway with a flight control system consisting of ***three independent channels***. Consider the system at the *critical instant* before takeoff:



Channel 1	**Take off**	**Accelerate Engine**
Channel 2	**Abort**	**Stop Engine**
Channel 3	Take off	*Stop Engine (Fault)*

Replica Determinism: Airplane on Takeoff

Consider an airplane that is taking off from a runway with a flight control system consisting of ***three independent channels***. Consider the system at the *critical instant* before takeoff:



Channel 1	**Take off**	**Accelerate Engine**
Channel 2	**Abort**	**Stop Engine**
Channel 3	Take off	*Stop Engine (Fault)*

Majority Take off *Stop Engine (Fault)*

Determinism of a Communication Channel

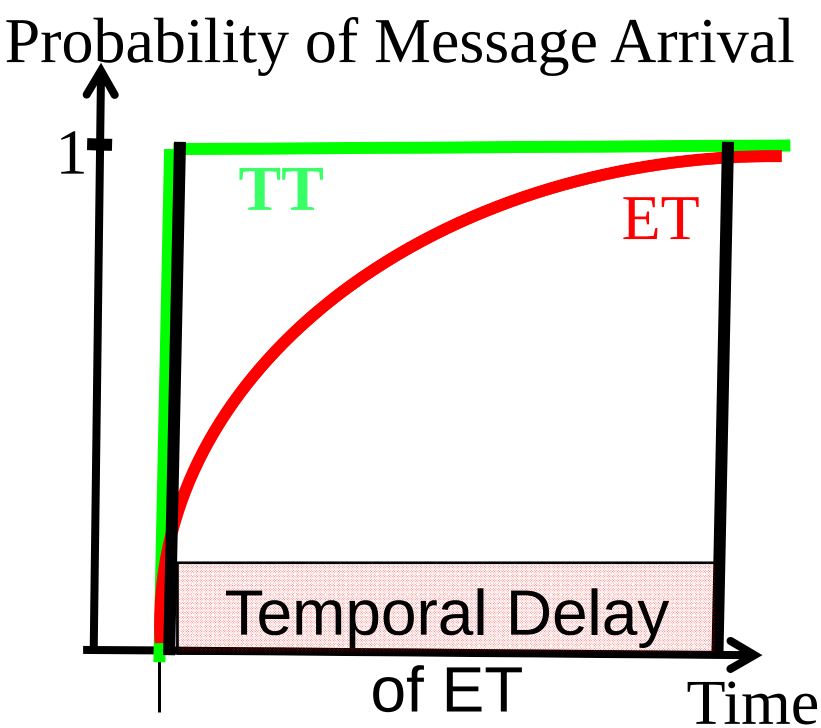
The behavior of a communication channel is called *deterministic* if (as seen from an omniscient external observer):

- A message is delivered before an *a priori* known instant (timeliness).
- The *receive order* of the messages is same as the *send order*. The *send order* among all messages is established by the *temporal order* of the *send instants* of the messages as observed by an omniscient observer.
- If the *send instants* of *n (n>1)* messages are the *same*, then an order of the *n* messages will be established in an *a priori* known manner.

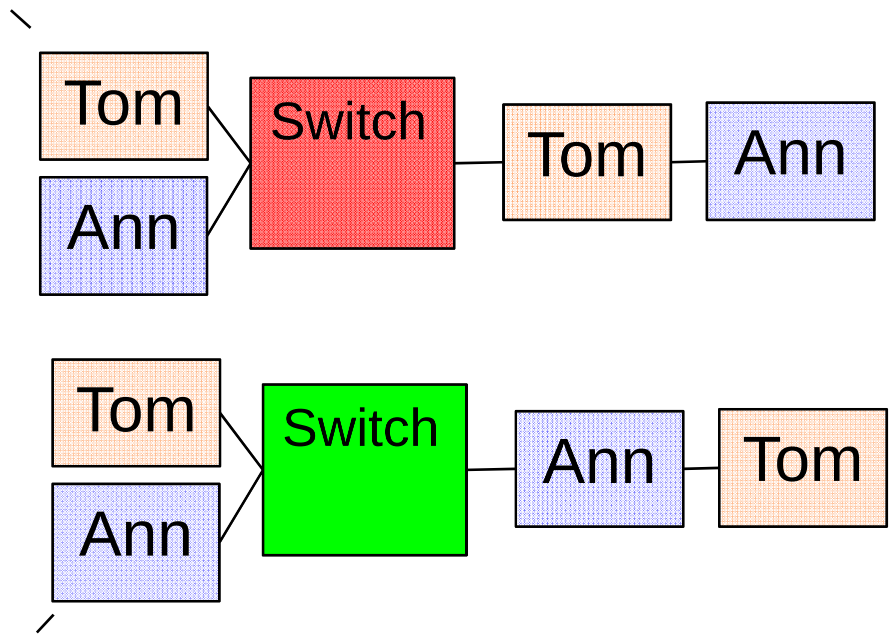
Two *independent* deterministic channels will deliver messages *always* in the same order before an *a priori* known instant.

Determinism: *Timeliness* and *Order*

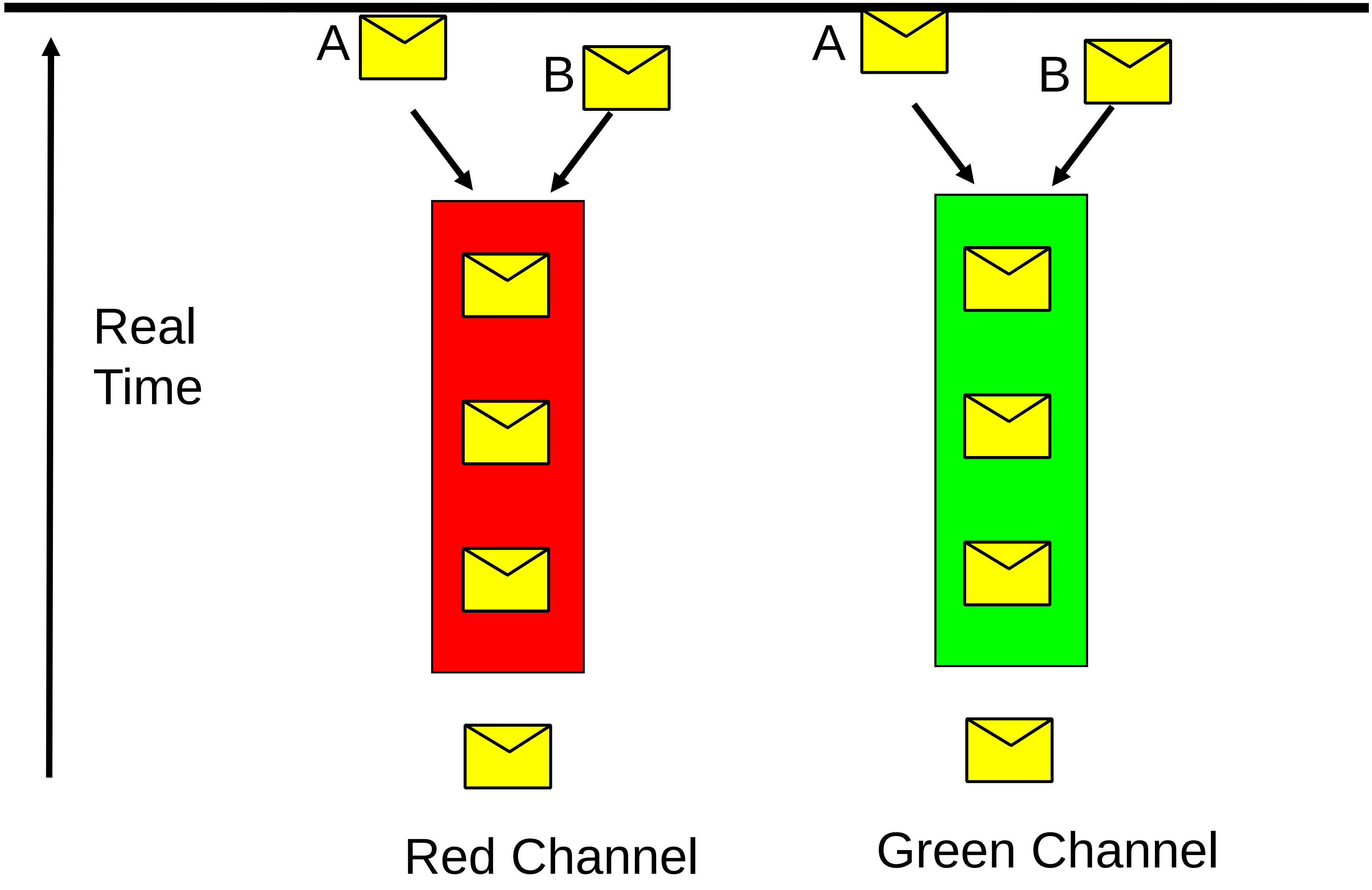
Timeliness



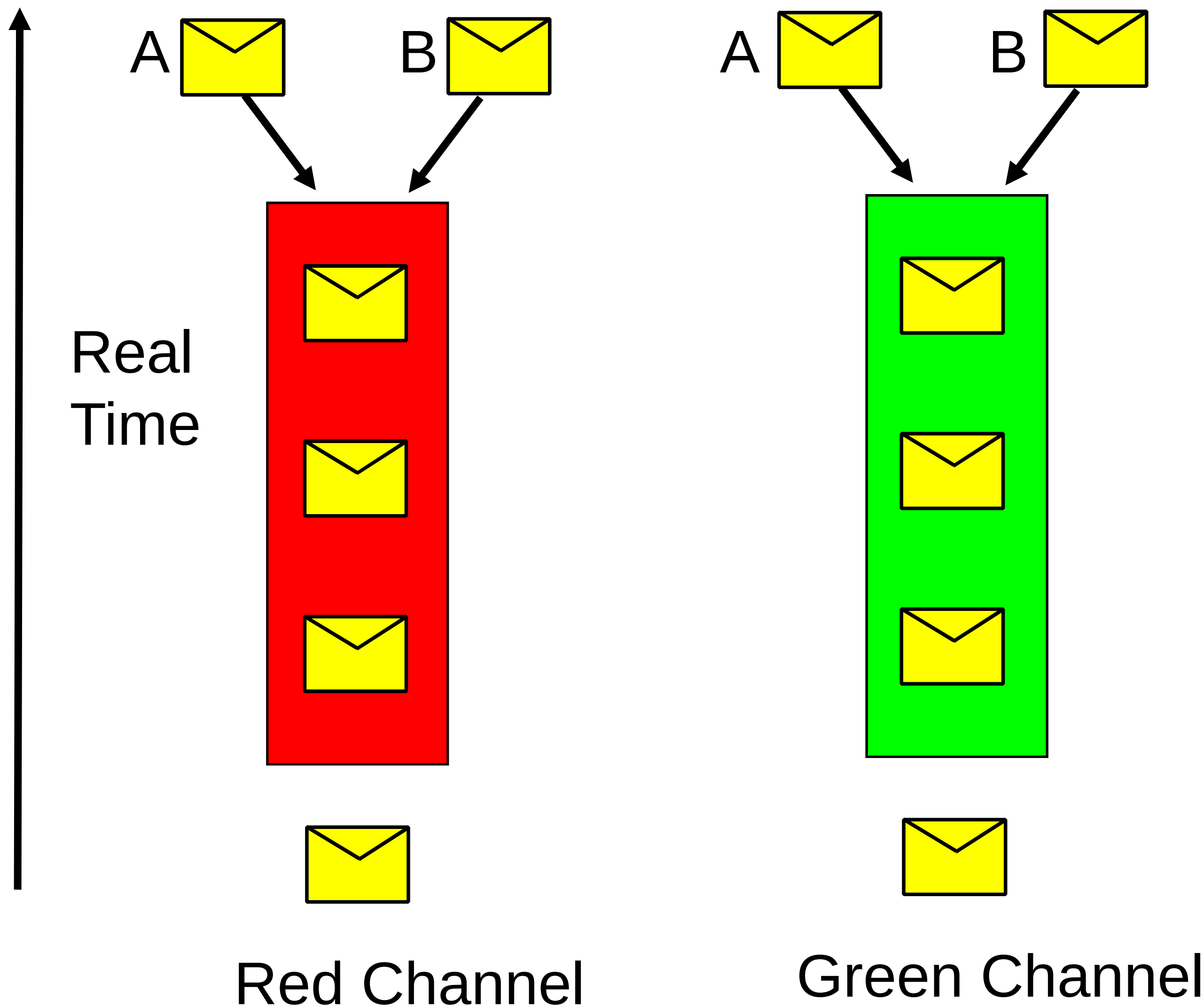
Consistent Temporal Order



Temporal Order is Obvious



Determinism: Simultaneity--Who Wins?



Determinisn:

If A wins on the red channel

then A must also win on the green channel

Handling of *Simultaneity*—A Fundamental Problem

In the hardware : *meta-stability*

In operating systems: *mutual exclusion*

In communication systems: *order of messages*

A two-step solution:

(i) Provide consistent view of simultaneity—distinguish between events that are in the sphere of control (SoC) of the system and events that are outside the SoC--*difficult*

(ii) Order simultaneous events according to some *a priori* established criterion--*easy*

Distribution of the Global Time

- In a global SoS environment: GPS with GPSDO
- In a wired local environment: IEEE 1588, TTE, TTP
- In wireless Systems: IEEE 802.11 E (WiFi), GPS?

Dependability of GPS

The following failure modes of the GPS signals must be considered:

- Natural Interference, e.g., Solar Wind, Geomagnetic Storm
- Jamming
- Spoofing

Countermeasures:

- Local Time Source with high precision, e.g., atomic clock, GPS disciplined oscillators
- Internal Clock Synchronization

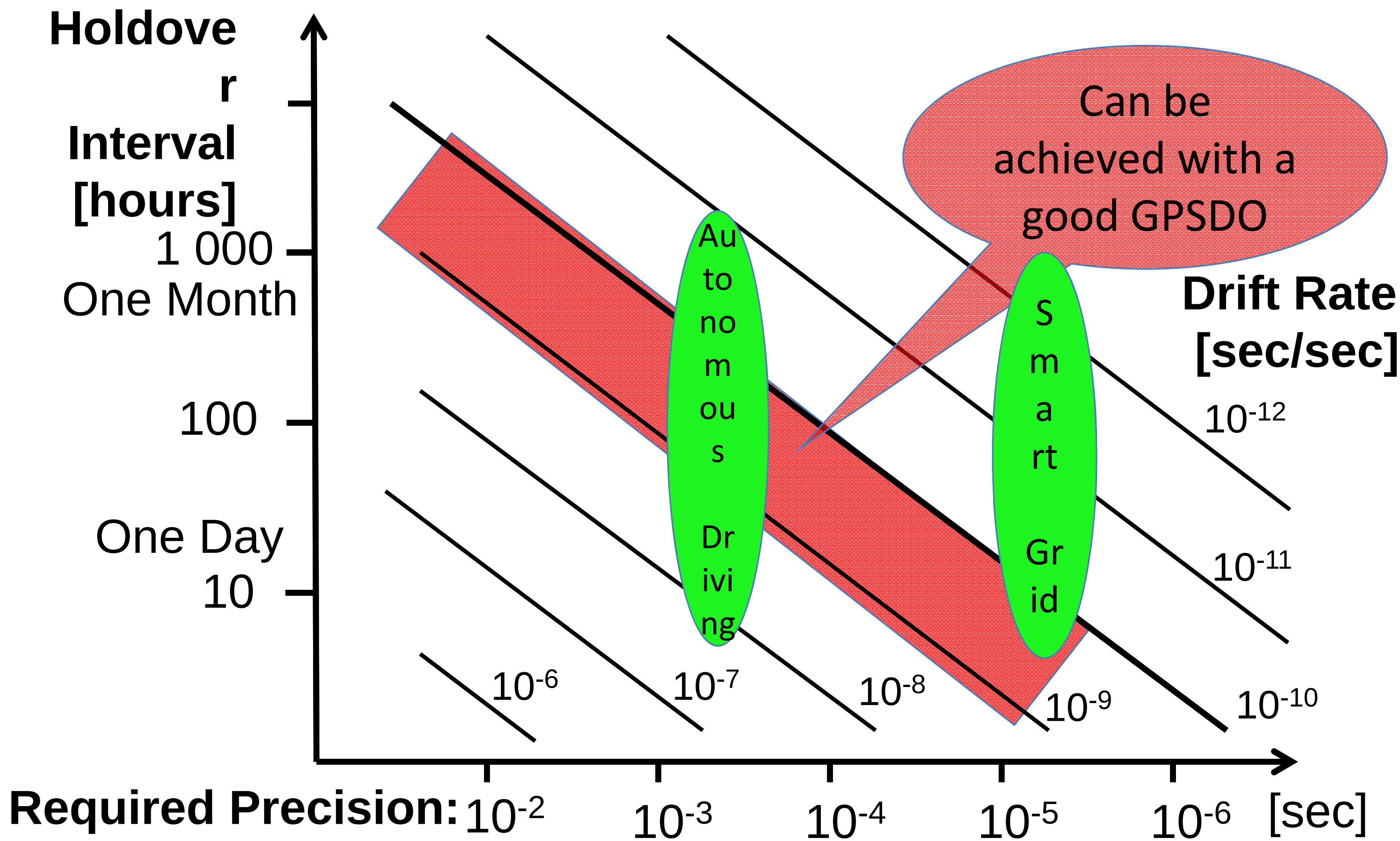
GPS Disciplined Oscillator (*GPSDO*)

A GPS disciplined oscillator is a clock where two operating modes are distinguished:

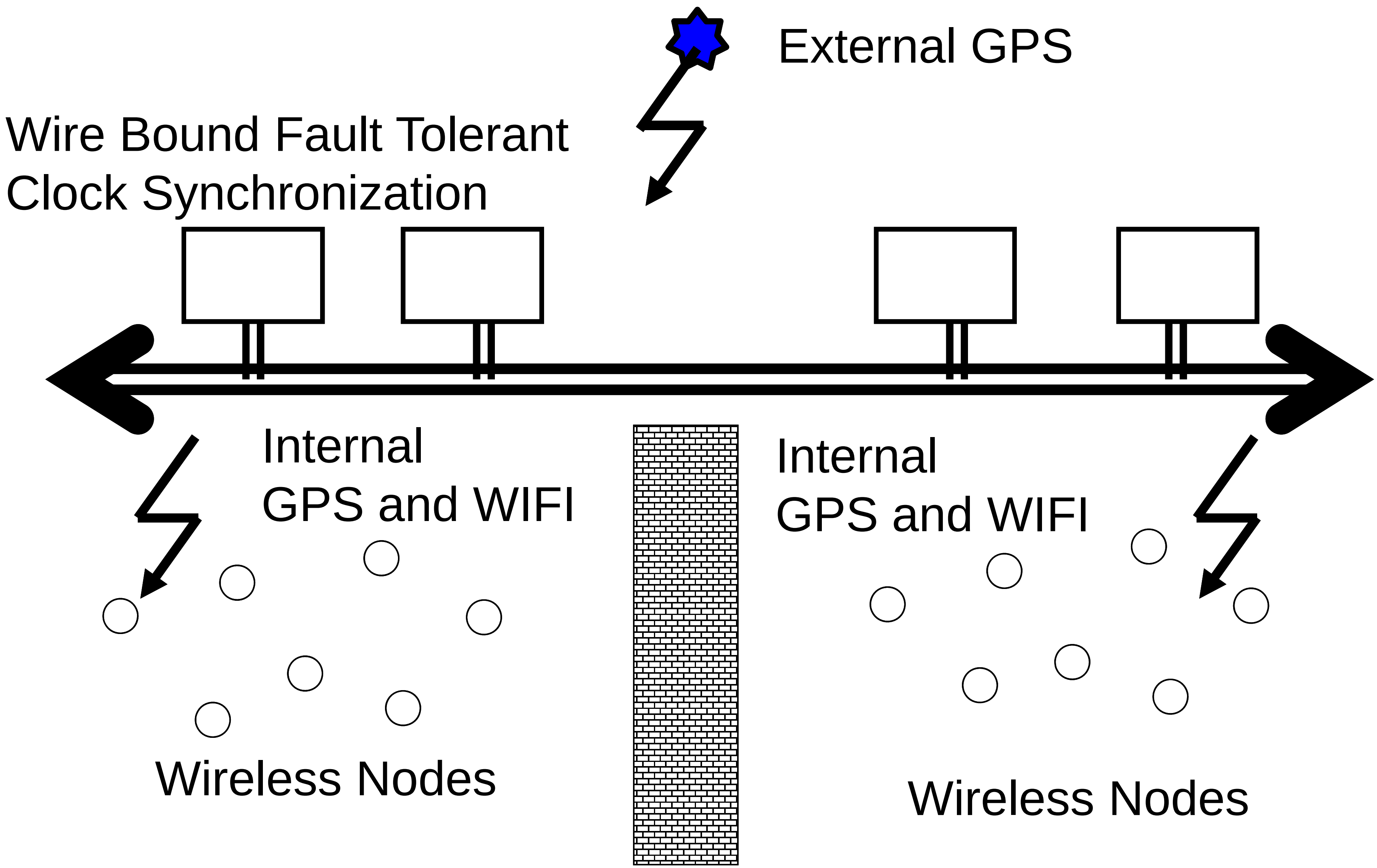
Learning Mode: In the learning mode that is realized as long as a valid GPS signal is received, the *state* and the *rate* of the clock are continually aligned with the GPS time signal.

Holdover Mode:: In the holdover mode that is realized when no valid GPS signal is received, the clock is free running, starting with the most recent GPS *state* and continuing with the adapted rate.

Holdover Interval of a Local Clock



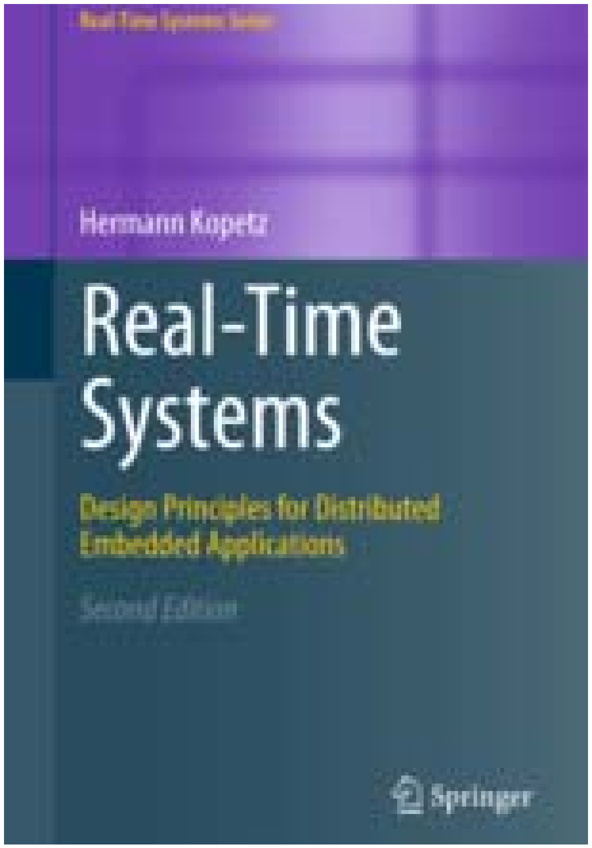
A Combined Approach . . .



Conclusions

- The desired emergent behavior of SoS comes about by the timely exchange of information and the coordinated actions of the constituent systems (CS).
- If a proper global time base of sufficient quality is provided in all CS, many of the coordination and synchronization problems are simplified.
- In an SoS, external synchronization, e.g., by GPS, is the preferred alternative.
- A sparse time-model brings about the required consistency of the cyber model at the price of reduced fidelity.

More Information



Background information can be found in the second revised edition of my book

Real-Time Systems—Design Principles for Distributed Embedded Applications

published by ***Springer Verlag*** on April 27 , 2011.