Time-Triggered Protocols for Safety-Critical Applications

Hermann Kopetz
TU Wien
March 21, 2001

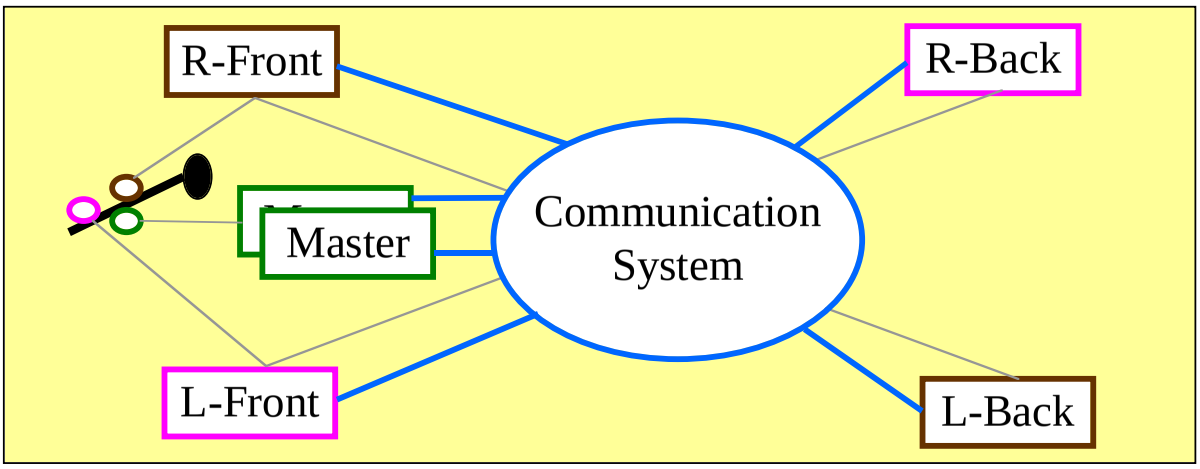
Outline

- Introduction
- State and Event Information
- Why Time-Triggered Communication?
- Example of TT Protocols
- Integration of ET and TT Services
- Conclusion

Safety Critical Applications

- Embedded Computer System is part of a larger system that performs a safety-critical service.
- Failure of the system can cause harm to human life or extensive financial loss.
- In most cases, tight interaction with the environment: real-time response of the computer system required.
- System must perform predictably, even in the case of a failure of a computer or the enclosing system.
- No single point of failure requires a distributed computer architecture.

Example: Brake-by-Wire System



Essential Characteristics of RT Systems

Physical time is a first order concept: There is only one physical time in the world and it makes a lot of sense to provide access to this physical time in all nodes of a distributed real-time system.

Time-bounded validity of real-time data: The validity of real-time data is invalidated by the progression of real-time.

Existence of deadlines: A real-time task must produce results before the deadline--a known instant on the timeline--is reached.

Inherent distribution: Smart sensors and actuators are nodes of a distributed real-time computer system.

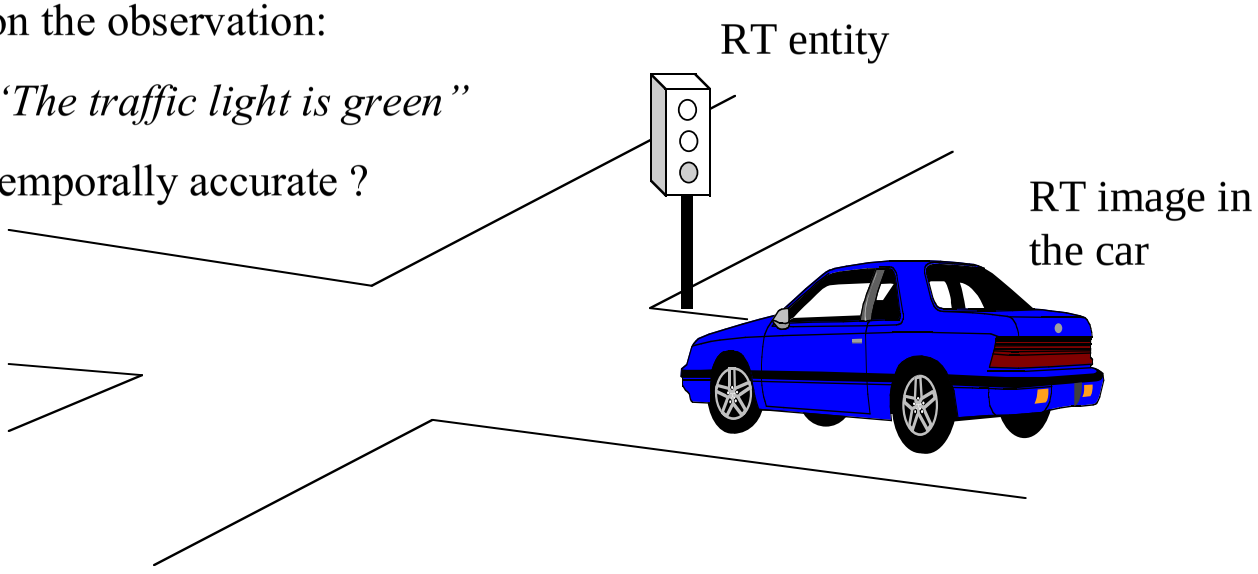
High dependability: Many real-time systems must continue to operate even after a component has failed.

Temporal Accuracy of Real-Time Information

How long is the RT image, based on the observation:

"The traffic light is green"

temporally accurate ?



If the correct value is used at the wrong time, its just as bad as the opposite.

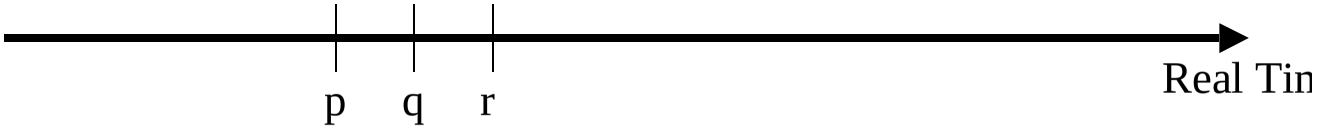
Model of Time (Newton)--Temporal Order

The continuum of real time can be modeled by a directed timeline consisting of an infinite set {T} of *instants* with the following properties:

(i) {T} is an ordered set, i.e., if p and q are any two instants, then either (1) p is simultaneous with q or (2) p precedes q or (3) q precedes p and these relations are mutually exclusive. We call the order of instants on the timeline the *temporal order*.

(ii) {T} is a dense set. This means that, if p≠r, there is at least one q between p and r.

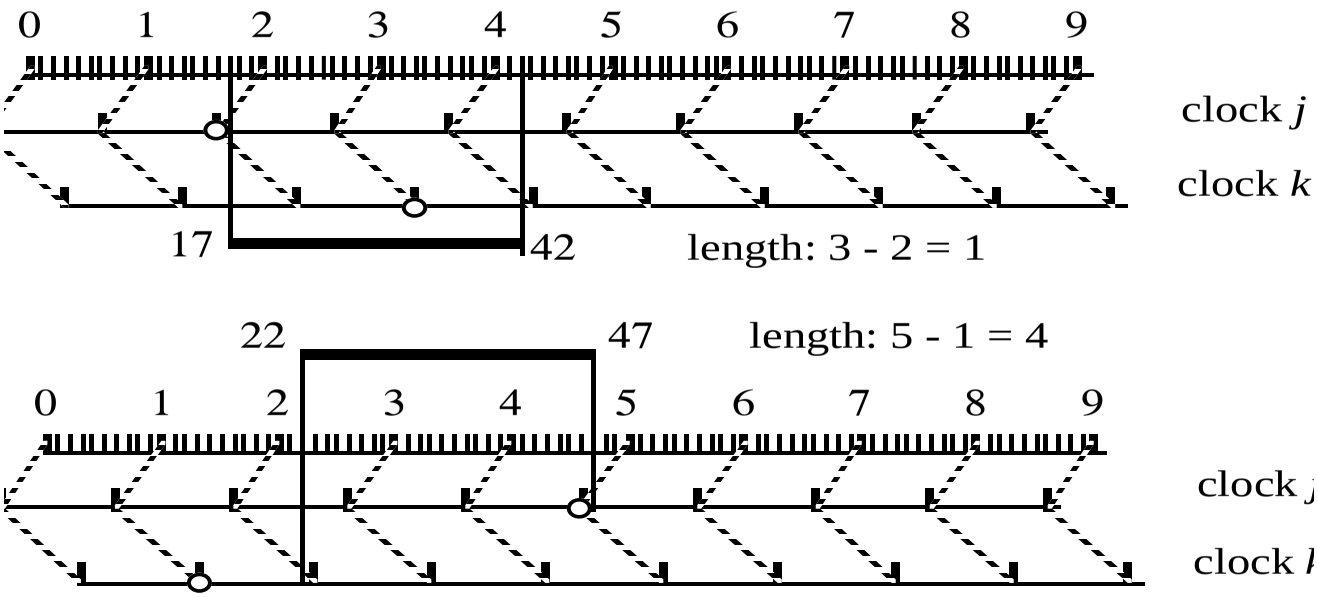
The order of instants on the timeline is called the *temporal order*.



Durations and Events

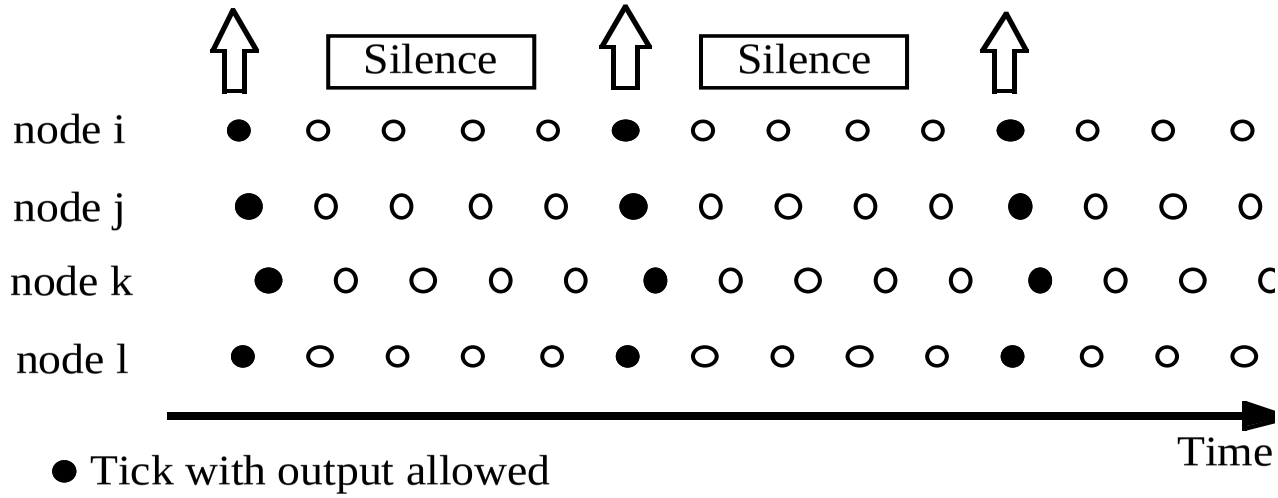
- A section of the time line is called a *duration*.
- An *event* is a happening at an instant of time.
- An event does not have a duration. If two events occur at an identical instant, then the two events are said to occur simultaneously.
- Instants are totally ordered; however, events are only partially ordered, since simultaneous events are not in the order relation.

Interval Measurement



It follows: $(d_{obs} - 2g) < d_{true} < (d_{obs} + 2g)$

Space/Time Lattice



- ● Tick with output allowed
- ○ Tick with output not allowed

Causal Order

Reichenbach [Rei57,p.145] defined *causality* by a mark method without reference to time: "If event e1 is a cause of event e2, then a small variation (a mark) in e1 is associated with small variation in e2, whereas small variations in e2 are not necessarily associated with small variations in e1."

Example: Suppose there are two events e1 and e2:

e1 Somebody enters a room.

e2 The telephone starts to ring.

Consider the following two cases

(i) e2 occurs after e1

(ii) e1 occurs after e2

Real Time (RT) Entity

A Real-Time (RT) Entity is a state variable of interest for the given purpose that changes its state as a function of real-time.

We distinguish between:

- Continuous RT Entities
- Discrete RT Entities

Examples of RT Entities:

- Flow in a Pipe (Continuous)
- Position of a Switch (Discrete)
- Setpoint selected by an Operator
- Intended Position of an Actuator

Observation

Information about the state of a RT-entity at a particular point in time is captured in the concept of an **observation**.

An *observation* is an atomic triple

Observation = <Name, Time, Value>

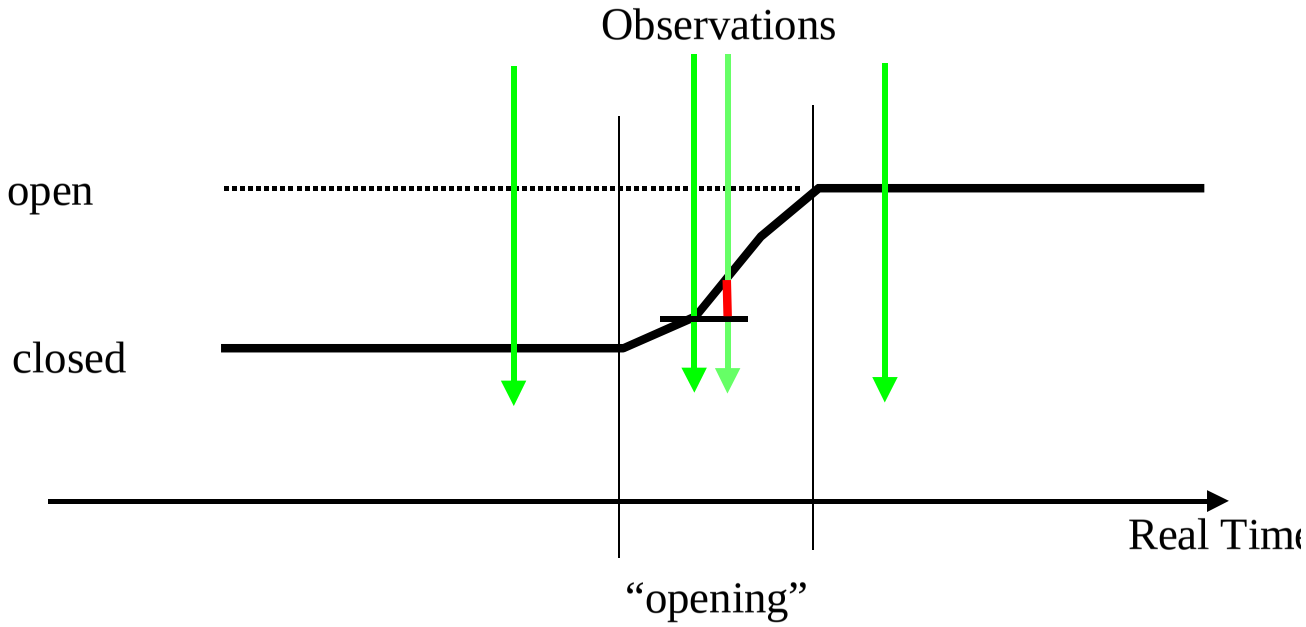
consisting of:

- The name of the RT-entity
- The point in real-time when the observation has been made
- The values of the RT-entity

Observations are transported in messages.

If the time of message arrival is taken as the time of observation, delaying a message changes the contained observation.

Observation of a Valve



State and Event Observation

An observation is a *state observation*, if the value of the observation contains the full or partial state of the RT-entity. The time of a state observation denotes the point in time when the RT-entity was sampled.

An observation is an *event observation*, if the value of the observation contains the difference between the "old state" (the last observed state) and the "new state". The time of the event information denotes the point in time of observation of the "new state".

What is the Difference?

	State	Event
Time of Observation	periodic	after event occurrence
Trigger of Observation	**Time**	**Event**
Content	Full state	Difference new - old
Required Semantics	**at-least once**	**exactly once**
Loss of observation	short blackout	loss of state synchronization
Idempotency	yes	no

Event Triggered (ET) vs. Time Triggered (TT)

A Real-Time system is *Event Triggered* (ET) if the control signals are derived solely from the occurrence of events, e.g.,

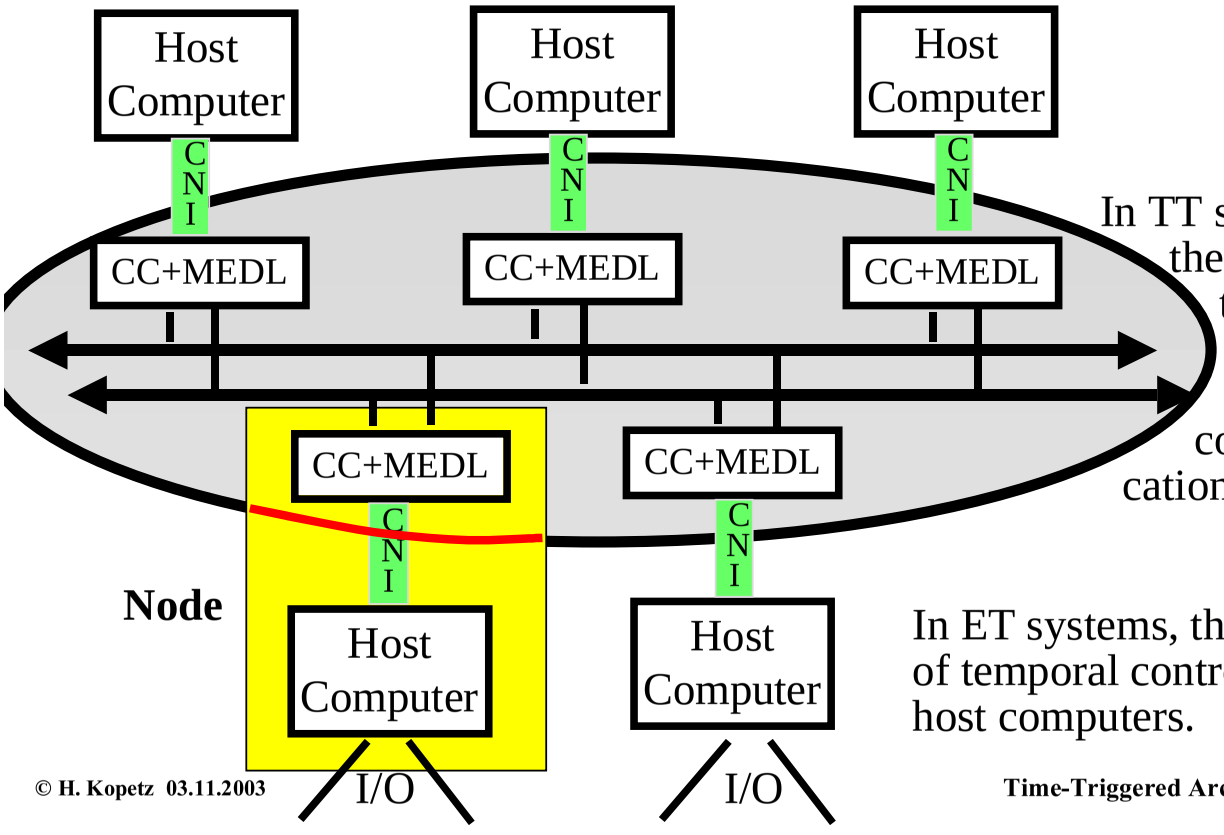
- termination of a task
- reception of a message
- an external interrupt

A Real-Time system is *Time Triggered* (TT) if the control signals, such as

- sending and receiving of messages
- recognition of an external state change

are derived solely from the progression of a (global) notion of time.

Global Interactions versus Local Processing



In TT systems the locus o tempora contrc is in the communi cation system

In ET systems, the locus of temporal control is in host computers.

Event Message versus State Message

Event Messages are event triggered:

- contain *event information*
- queued and consumed (exactly-once semantics)
- external control outside the communication system in the software in the host computer of a node.

State Messages are time triggered:

- contain *state information*
- atomic update in place by single sender, not consumed on reading, many readers
- sent periodically, autonomous control within communication system

State messages are appropriate for control applications.

Event Message versus State Message I

Characteristic	Event Message	State Message
Example of message contents	"Valve has closed by 5 degrees"	"Valve stands at 60 degrees"
Contents of data field	event information	state information
Instant of sending	After event occurrence	Periodically at *a priori* known points in time.
Temporal control	Interrupt caused by event occurrence	sampling, caused by th progression of time
Handling at receiver	queued and consumed on reading	new version replaces previous version, not consumed on reading
Semantics at receiver	Exactly once	At least once

Event Message versus State Message II

Characteristic	Event Message	State Message
Idempotence [Kopetz97, p.110]	no	yes
Consequences of message loss	Loss of state synchronization between sender and receiver	Unavailability of current state information for a sampling interval.
Typical communication protocol	Positive Acknowledgment or Retransmission (PAR)	Unidirectional datagram
Typical communication topology	Point to point	Multicast
Load on communication system	Depends on number of event occurrences	Constant

In Non-Real-Time Systems

- The interest is on state changes, i.e., events.
- Timely information delivery is not an issue, since time is not a key resource.
- Temporal composability is not an issue.
- Fault tolerance is achieved by checkpoint restart, not by active redundancy, which requires replica determinism.

In the “non real-time” world, event-triggered protocols, many of them non-deterministic (e.g., ETHERNET) are widely deployed.

Proactive Fault Analysis in Safety Critical Systems

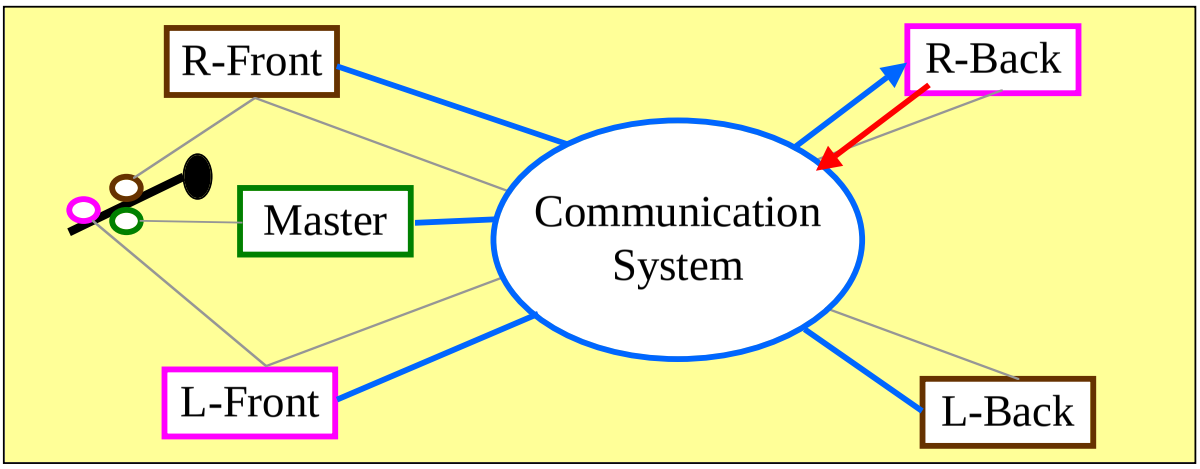
During the design of a safety critical system, all "thinkable" failure scenarios must be rigorously analyzed.

For example, in the aerospace community the following "checks" must be done:

- Any physical unit (chip) can fail in an arbitrary failure mode with a probability of 10^{-6}/hour
- Any matter in a physical volume of defined extension can be destroyed (e.g., by an explosion)--spatial proximity faults.
- . . . . . .

Total system safety must be better than 10^{-9}/hour.

Outgoing Link Failure--Membership



How to achieve consistency if a node has an outgoing link failure?
Only membership solves the problem!

Membership in ET versus TT

Every node must inform every other node about its local view of the "health state" of the other nodes--*and this in time*.

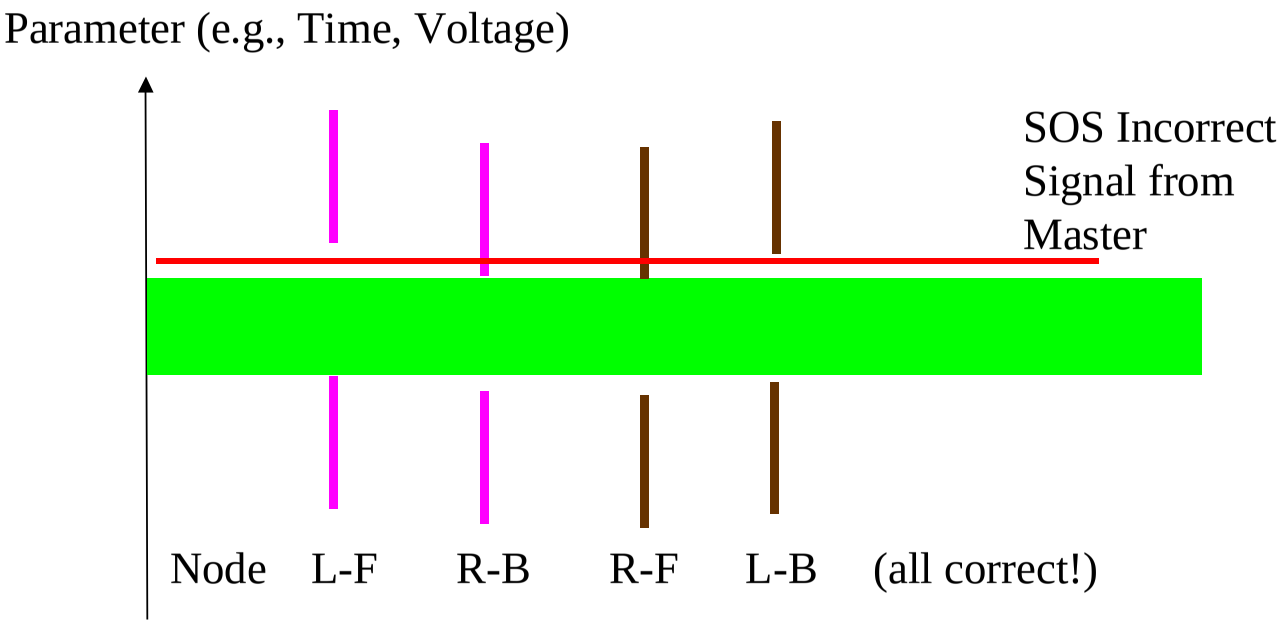
Event Triggered (e.g, CAN)

- **Membership difficult--message showers**
- Message arrival determined by the occurrence of events *unpredictable*
- Large Jitter
- No precise temporal specification of interfaces

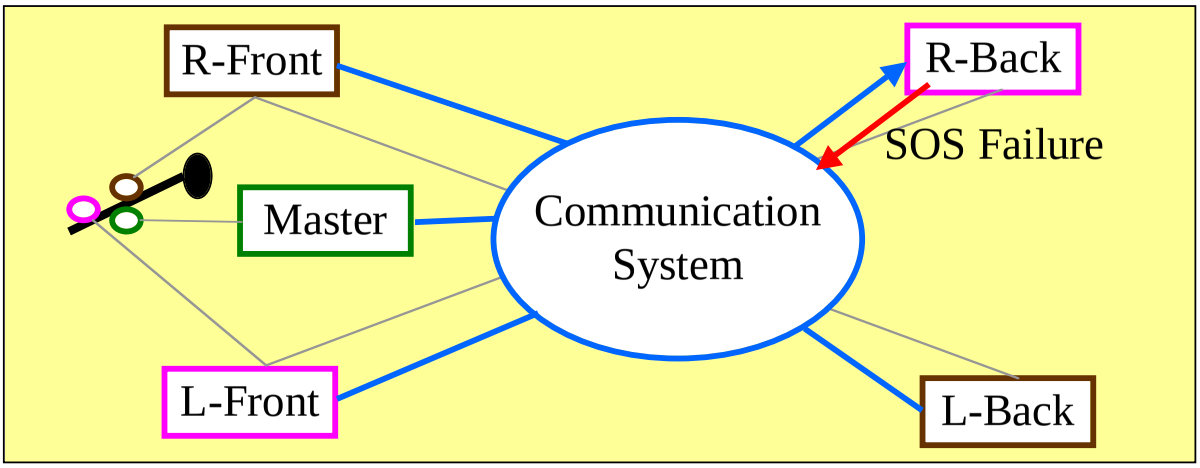
Time Triggered (e.g., TTP)

- **Membership easy--can be performed indirectly**
- Message arrival determined by the progression of time *predictable*
- Minimal Jitter.
- Interfaces are temporal firewalls.

Slightly-off-specification (SOS) Faults



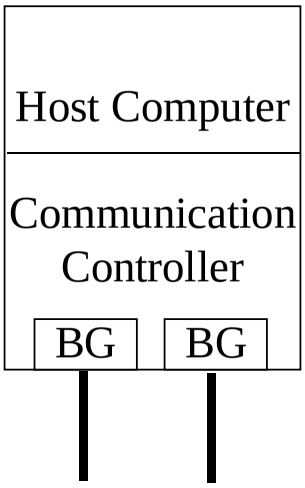
Outgoing SOS Link Failure



Replicated channels will not mask SOS failures if they are caused by the common clock or the common power supply of both channels.

Node Design

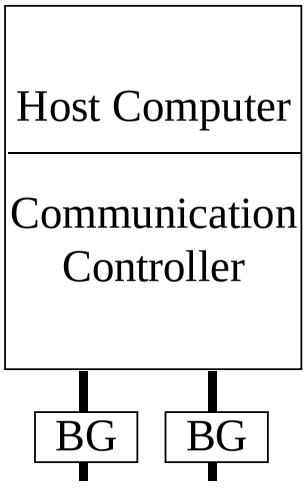
Previous Design



How to handle SOS faults if BG and node depend on the same clock and the same power?

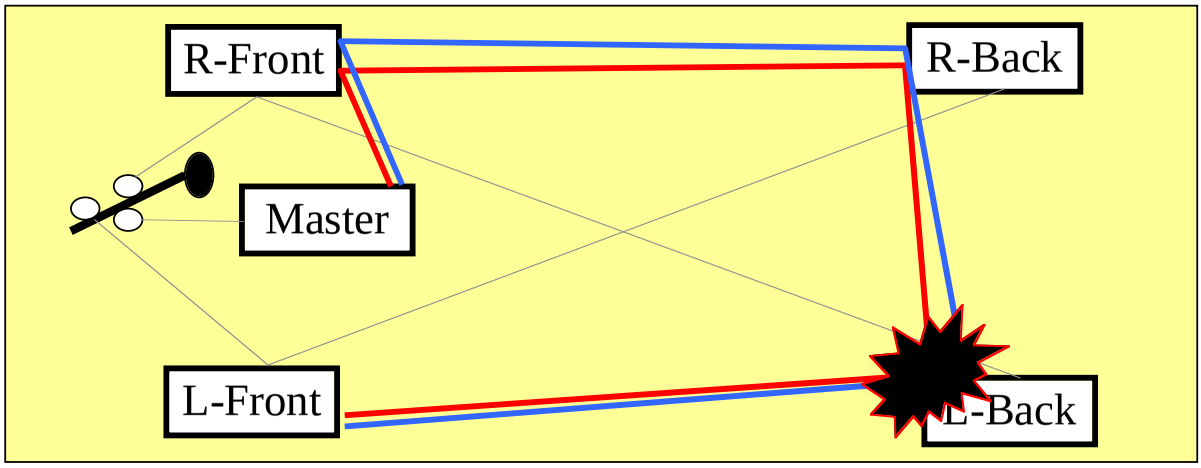
BG: Bus Guardian

Alternate Design



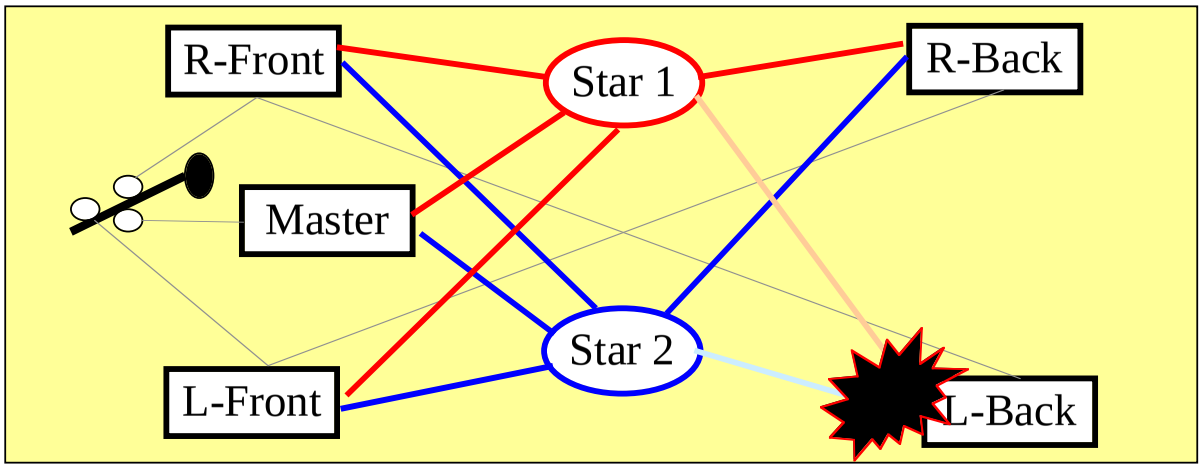
BG independent with its own clock and power supply, performs signal reshaping

Spatial Proximity Faults in Bus Systems



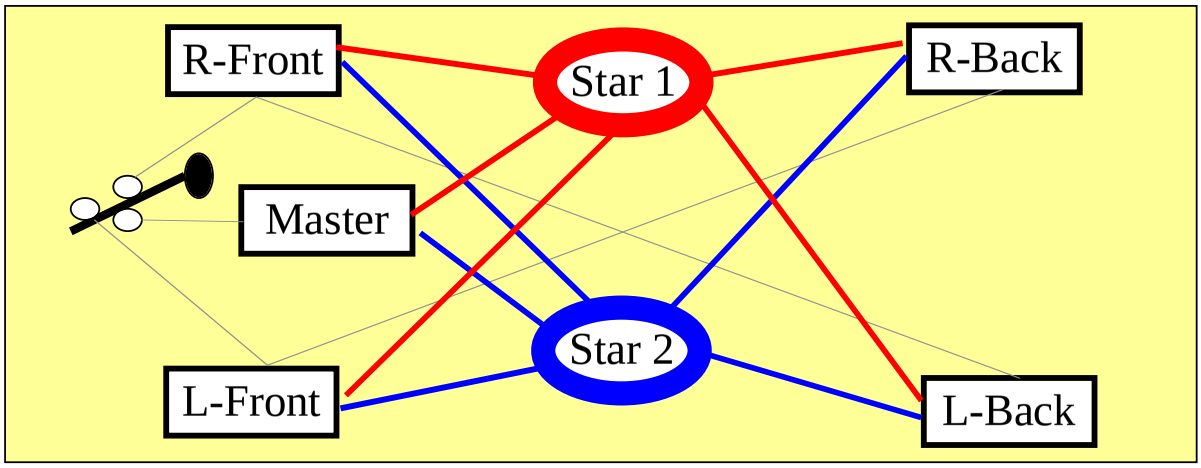
At every node, both busses must come into close physical proximity--creating many single points of (physical) failure.

Replicated Stars avoid Single Point of Failure



No defined volume of space becomes a single fault containment region, that can be a cause of total system failure.

Star with Bus Guardian handles both Fault Classes



An architecture with properly designed intelligent star couplers with signal reshaping tolerates **both**, SOS faults and physical proximity faults, with reasonable costs.

Some Time-Triggered Protocols

	Year	Chips	FT	Memb.	SOS	Spatial
SAFEbus	1992	1994	yes	no	yes	no
TTP/C	1994	1998	yes	yes	yes	yes
TTP/A	1997	*1997*	no	yes	no	no
LIN	1999	*1999*	no	no	no	no
TT-CAN	1999	2002?	no	no	no	no

SAFEBus

- Developed by Honeywell at the beginning of the 90ties for application in the Boeing 777 aircraft
- Standardized by ARINC (ARINC 659)
- Time-triggered protocol
- Designed as a backplane bus, consisting of two selfchecking buses.
- Only bit-by-bit identical data is written into the memory
- Space and time determinism are supported.

SAFEBus Principles:

- "If a system design does not built in time determinism, a function can be certified only after all possible combinations of events , including all possible combinations of failures of all functions, have been considered".
- "Any protocol that includes a destination memory address is a space-partitioning problem".
- "Any protocol that uses arbitration cannot be made time-deterministic".

Source: Driscoll, 1994

TTP/C Protocol Services

The Time-Triggered Protocol (TTP), connecting the nodes of the system, is at the core of the Time-Triggered Architecture. It provides the following services:

- Predictable communication with small latency an minimal jitter
- Fault-tolerant clock synchronisation
- Composability by full specification of the temporal properties of the interfaces.
- timely membership service (fast error detection)
- replica determinism
- replicated communication channels (support of fault- tolerance)
- good data efficiency

TTP/C Silicon

TTP/C is an open technology. The TTP/C specification is on the Web. More than 2000 companies have downloaded the TTP/C specification

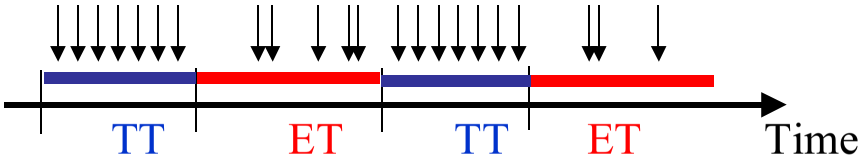
- TTP silicon, supporting 2 Mbits/s is available since 1998.
- A TTP/C chip which supports up to 25 Mbit/s is expected to be available before the end of this year.
- A Gigabit implementation of TTP/C is being investigated in a research project.
- TTP/C design models are made available to semiconductor companies in order to integrate TTP/C on system chips.

From the point of view of fault containment, the TTA architecture has been designed so that it can be implemented with a minimal number of chip packages.

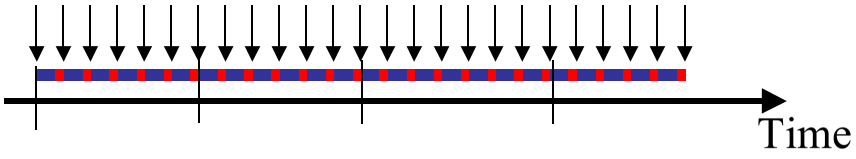
Integration of TT and ET Services

Two possible alternatives

(i) **Parallel**: Time Axes is divided into two parallel windows, where one window is used for TT, the other for ET, Two media access protocols needed, one TT, the other ET



(ii) **Layered**: ET service is implemented on top of a TT protocol Single time triggered access media access protocol.



Tradeoffs between Parallel and Layered ET

	Parallel ET	Layered ET
System wide band-width sharing possible	yes	no
Host interruptions	unknown	known
Temporal composability	no	yes
Protocol complexity	larger (2 protocols)	smaller

ET Services in TTP

Data-elements in a message are classified according to their contents:

- Event information--event semantics or
- State information--state semantics.

State information is stored in dual ported RAM.

Event information is presented according to the rules of a selected event protocol

- CAN
- TCP/IP

Basic TTP/C protocol is unchanged, maintaining the composability of the architecture.

Example of ET Integration

TTP/C system with 10 Mbit/sec transmission speed

10 nodes, Message length 400 bits (40 μsec), IFG 10 μsec,

7 bytes/message (about 15 % of bandwidth allocated for ET traffic)

CAN Message length: 14 bytes, i.e,

- One CAN message/(node.msec.)
- Total 10 000 CAN messages/second (corresponds to 1120 kbits/sec CAN channel)
- 85 % of the bandwidth is available for TT traffic.
- Scaleable to higher speeds

Multi-level Safety

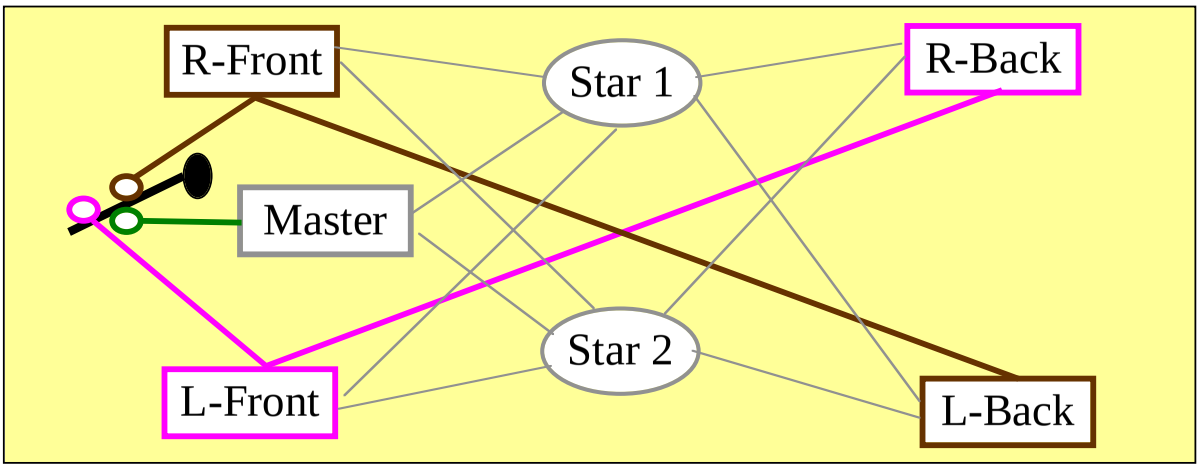
In safety critical systems, a multi-level approach to safety is often required:

- Requires levels of fault hypothesis
- Remaining safety margin important
- Design diversity with different implementation technologies should be considered

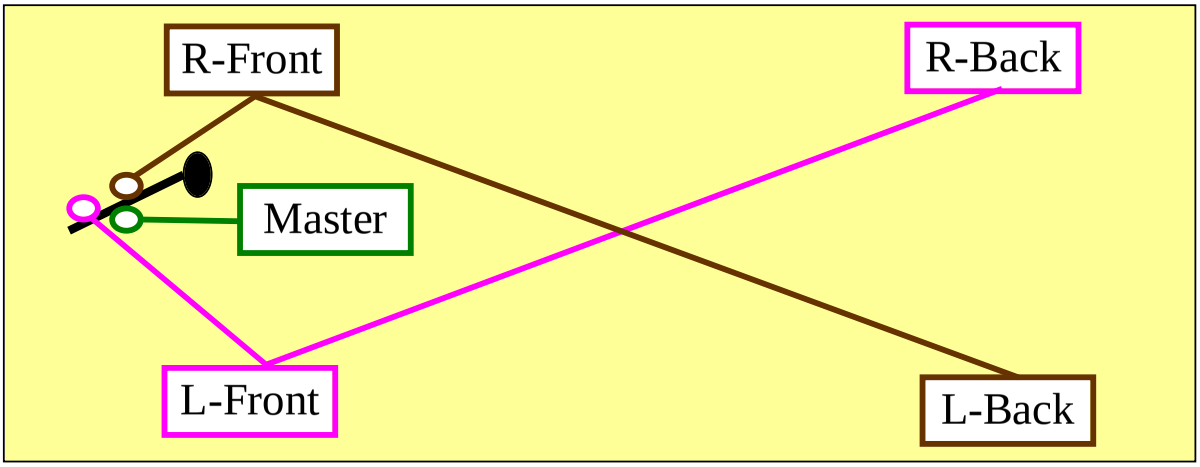
Fault Scenarios

- Level 1: Transient single node failure: Single Actuator frozen, node recovers within 10 msec recovery time
- Level 2: Permanent single node failure: Brake force redistributed to remaining three nodes
- Level 3: Transient communication system failure: All actuators frozen for node recovery time of 10 msec.
- Level 4: Permanent communication system failure: Braking system partitions into two independent diagonal braking subsystems.

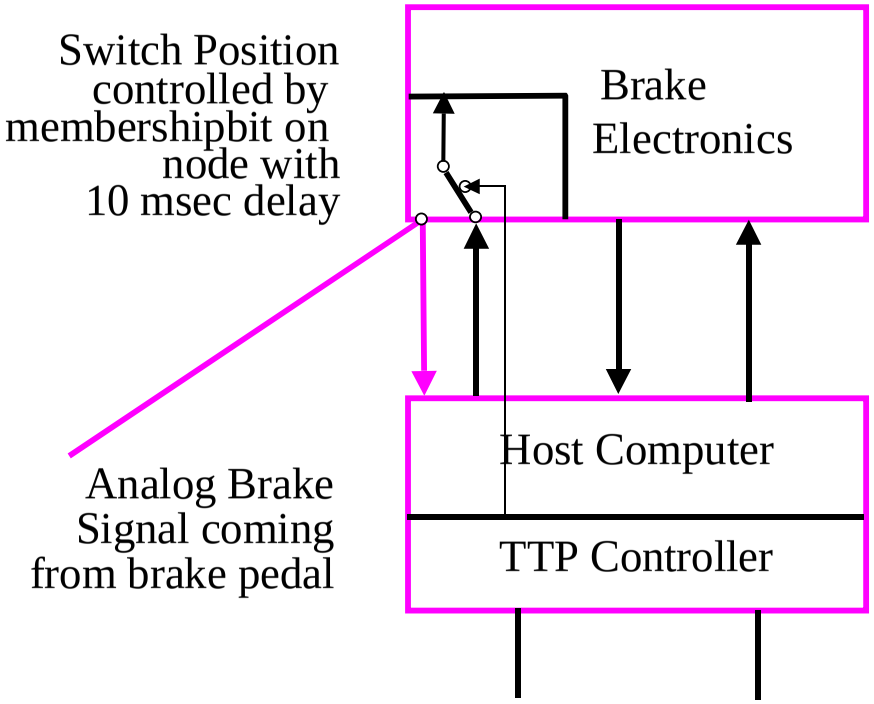
Total Loss of Digital Communication



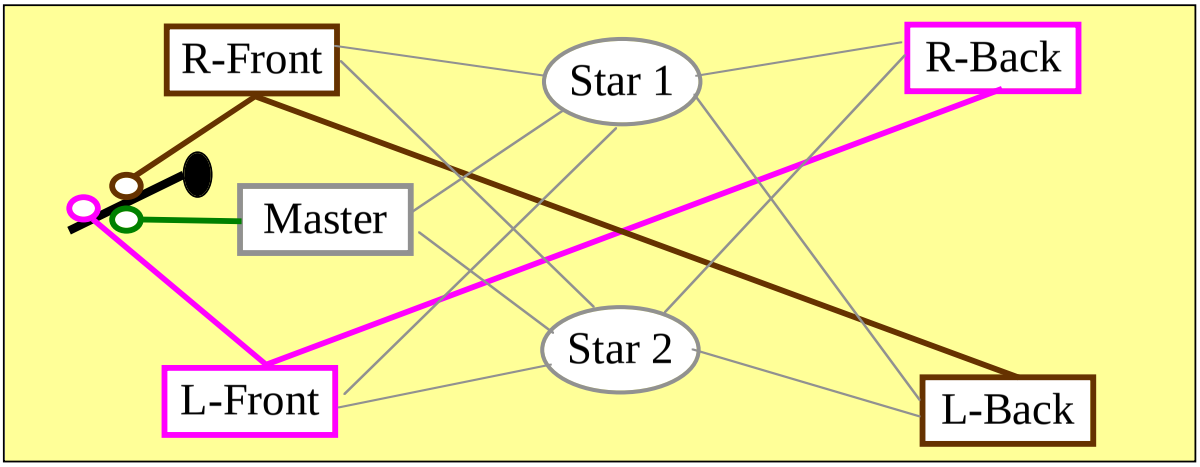
Sensor Interface



Wheel Computer Interface



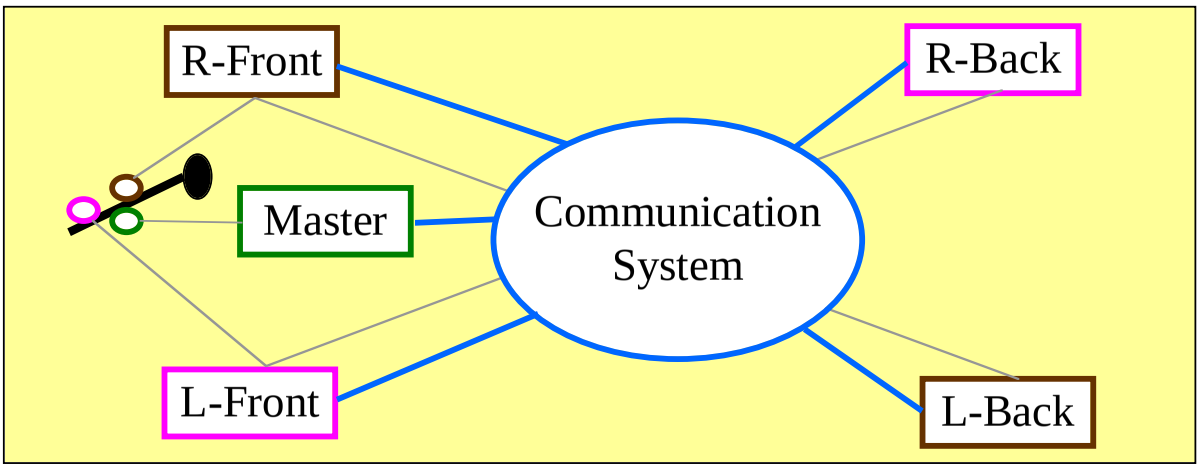
Total Loss of Digital Communication



Conclusion

- The time-triggered architecture with TTP/C as the main protocol is a mature architecture for the implementation of high-dependability systems in different application domains (automotive, aerospace, industrial electronics).
- The extensions to cover SOS faults and spatial proximity faults required no change to the TTP/C protocol.
- The standardisation of the TTA interfaces by the OMG and the access of TTA data by CORBA opens new avenues to interoperability on a world-wide scale.

Example: Brake-by-Wire System



Membership Service: Every node knows consistently (within a known small *temporal delay*) who is present and who is absent--**requires time awareness**.