Networks and Fieldbuses for Real-Time Embedded Systems

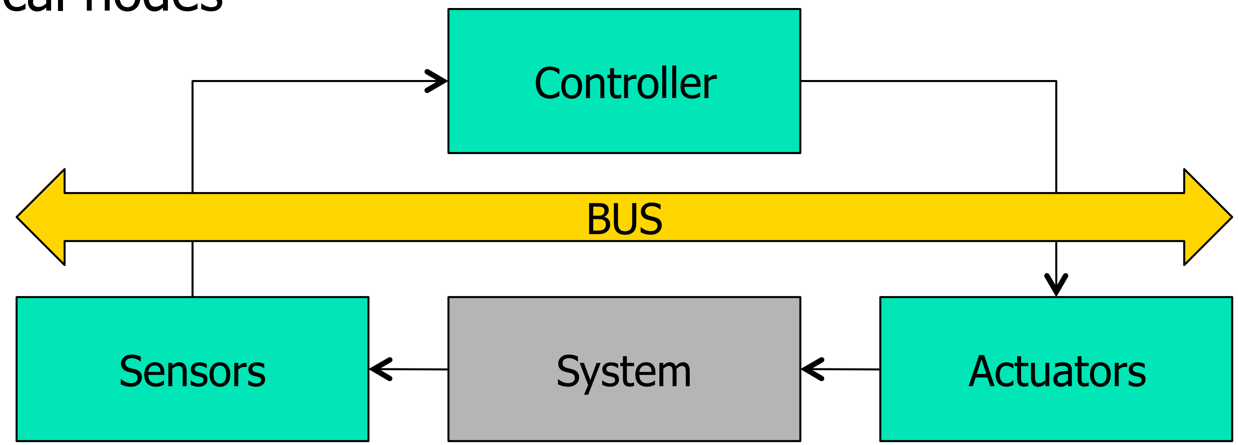
Laurent Pautet

Laurent.Pautet@enst.fr

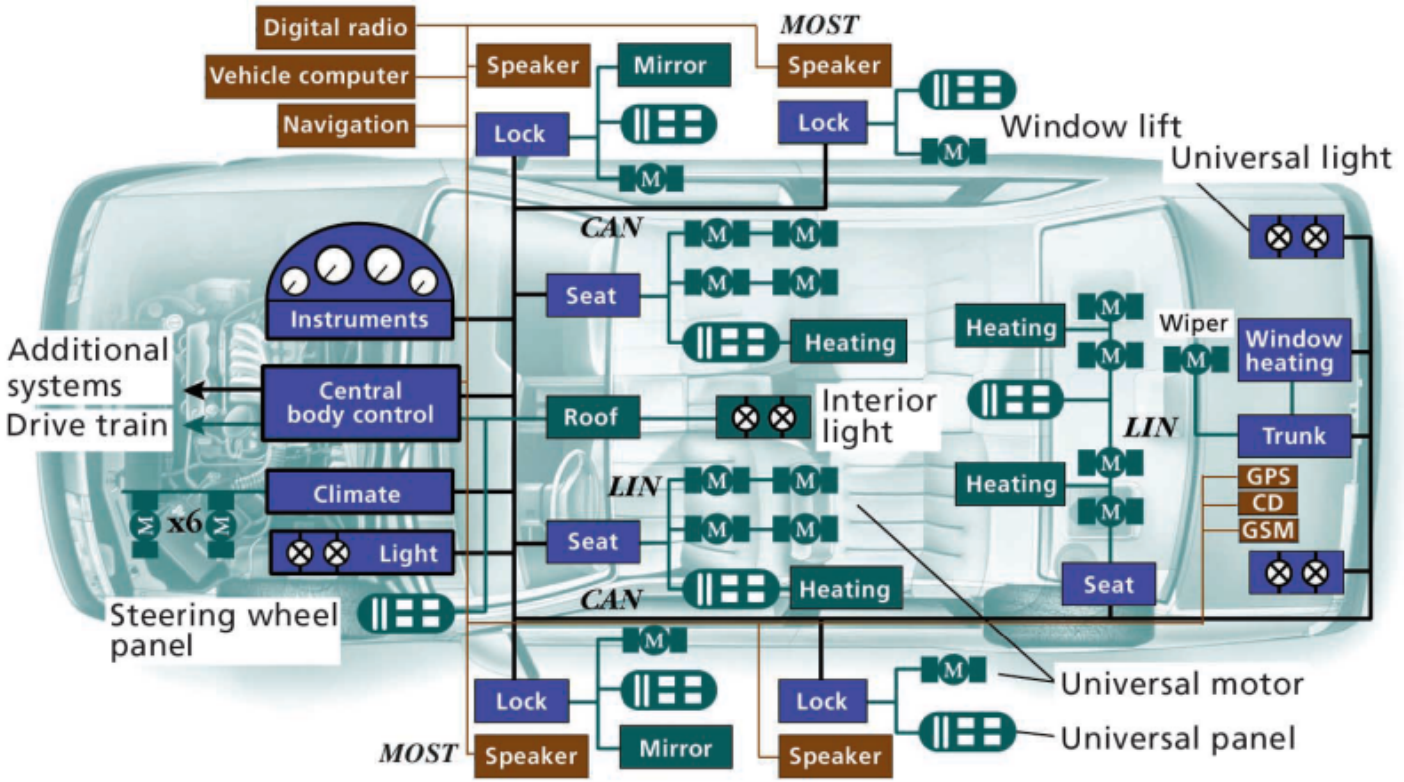
Version 1.0

Context

- The real-time embedded systems are often distributed (actors, sensors, remote computations).
  - Cyber Physical Systems
  - Transport systems
  - Multimedia systems
- Nodes are organized around control loops communicating through buses or networks
- Bus or network become critical resources as they are shared by critical nodes



Architecture of an automotive architecture



CAN Controller area network
GPS Global Positioning System
GSM Global System for Mobile Communications
LIN Local interconnect network
MOST Media-oriented systems transport

Constraints and Services

- The services to be provided are
  - Efficient small data transmissions
    - Transmission of simple sensors and actuators
  - Periodic transmissions controlled in time
    - Small periods, low latencies, low jitters
  - Aperiodic transmissions
    - Fast transmission of alarms
  - Non-real-time transmission of large data
    - Logging, images
  - Broadcast type transmissions

Interfaces, Messages, Transactions

- The bus or network consists of
  - Hardware infrastructure
  - Software layers
  - Communication interfaces (CNI)
- The message contains the data transmitted
  - But also contains control information
  - Involves a transaction or sequence of actions
- The efficiency of the network depends on the ratio between
  - The actual data and the induced overhead

Temporal and functional behavior

- Sources of temporal variations
  - Fixed deadlines (access time to the medium)
  - Variable delays (medium size)
  - Regulation buffer (flow problems)
  - Loss of message (unreliable medium)
- Types of messages
  - Events are ordered into a queue and removed after reading
  - The states or shared data are read several times and overwritten by the next state or data

Architecture

- Network architectures (like systems) are driven by event or time
- By event
  - Example: the speed has been increased by + dv
  - We must not lose the events
  - Not very deterministic, more flexible
- By time
  - Example: at date t, the speed is v
  - You have to support a large flow of data
  - More deterministic, less flexible

ISO Model

- ISO layers (traditional network)
  - Application (offered service, FTP, HTTP ...)
  - Presentation (data processing, compression ...)
  - Session (control, sequence of actions ...)
  - Transport (data transfer, TCP, UDP, ...)
  - Network (routing, logical address, …)
  - Data Link (medium access, collision, arbitration, ...)
  - Physical (physical transmission, copper, fiber, ...)

ISO Model Issues

- Each layer adds overhead due to control information (header)
- The ISO model aims at very general applications while real-time embedded systems are dedicated applications
- Closed set of applications and data : is a representation layer required ?
- Transmission in a restricted domain : is a network layer required ?

Light ISO Model

- For real-time requirements, all the overheads introduced by all these layers must be avoided.
- The Applications layer interacts directly with the Link layer which becomes the Communication Network Interface (CNI)
- These networks are then called Fieldbuses

Physical Layer

- The remaining 2 layers vary according to the domains and the applications but basically
  - The Physical layer
  - The Data Link layer
- The Physical layer
  - Topology in ring, bus, tree, star, ...
  - Copper, fiber, radio, infra-red equipment ...
  - Different characteristics (flow, error, ...)
  - Industrial constraints (price, temperature, ...)

Data Link Layer (1/2)

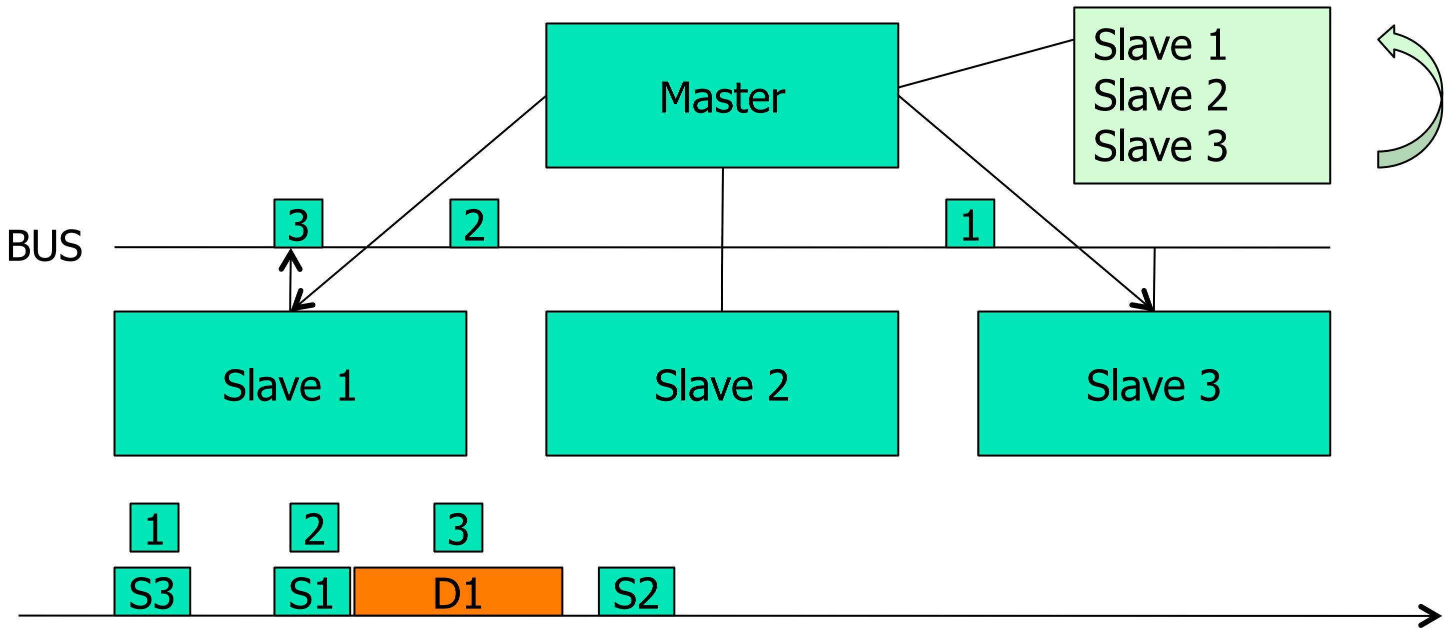
- The address of the receiver or transmitter is determined and optimised with the context
  - Receiver identifier known by design
  - Receiver known from date of transmission
  - Receiver known by the nature of the data received
- Data link control
  - Send with immediate acknowledgment
  - Send without acknowledgment
  - Communication by connection (ie TCP vs UDP)
  - ... depending performance needs

Data Link Layer (2/2)

- Transmission error check
  - Error correction code
  - Repetition on error signaled by the receiver
  - Repetition on lack of acknowledgment by sender
- Medium Access Control (MAC)
  - Master / Slave
  - Token ring
  - Time Division
  - Collision arbitration

MAC: Master / Slave

- A master invites a slave to send its data during an interval of time
- The slave uses the invitation to send data
- 1 message sent => +1 control message



Bus FIP

- Network for factory
- Bus or star topology
- Variable or message oriented
- Producer / consumer pattern
- The master sends a request to broadcast a variable
- The targetedd slave emits the value of the variable
- The other slaves read the value of the variable
- Communication table of periodic variables depending on their period and communication duration
- Send aperiodic requests for variables during idle time on the bus

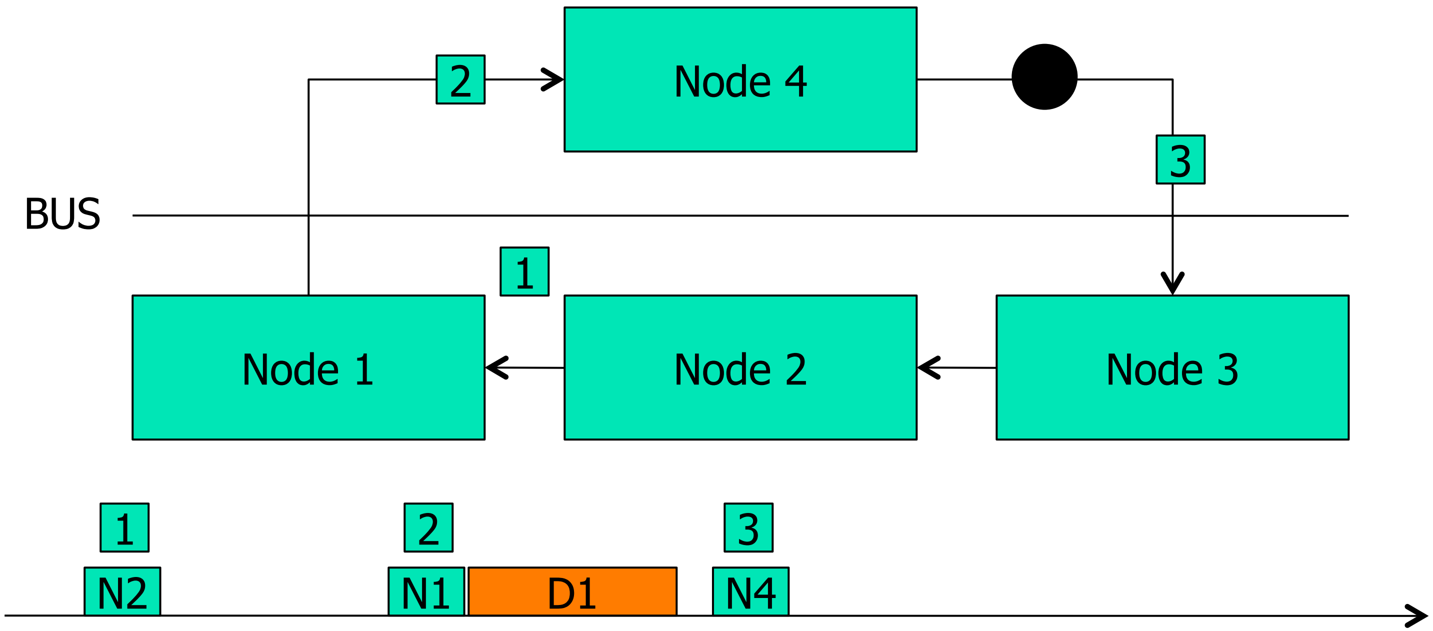
FIP Bus

Message	**Budget**	**Period**
M1	5ms	10ms
M2	3ms	20ms
M3	1ms	40ms

00ms - 10ms	M1	M1	M1	M1	M1	M2	M2	M2	M3	
10ms - 20ms	M1	M1	M1	M1	M1					
20ms – 30ms	M1	M1	M1	M1	M1	M2	M2	M2		
30ms – 40ms	M1	M1	M1	M1	M1					

MAC: Token Ring

- The token is used to send data for a period of time
- The token circulates on a ring between the nodes
- The node must release the token after a deadline

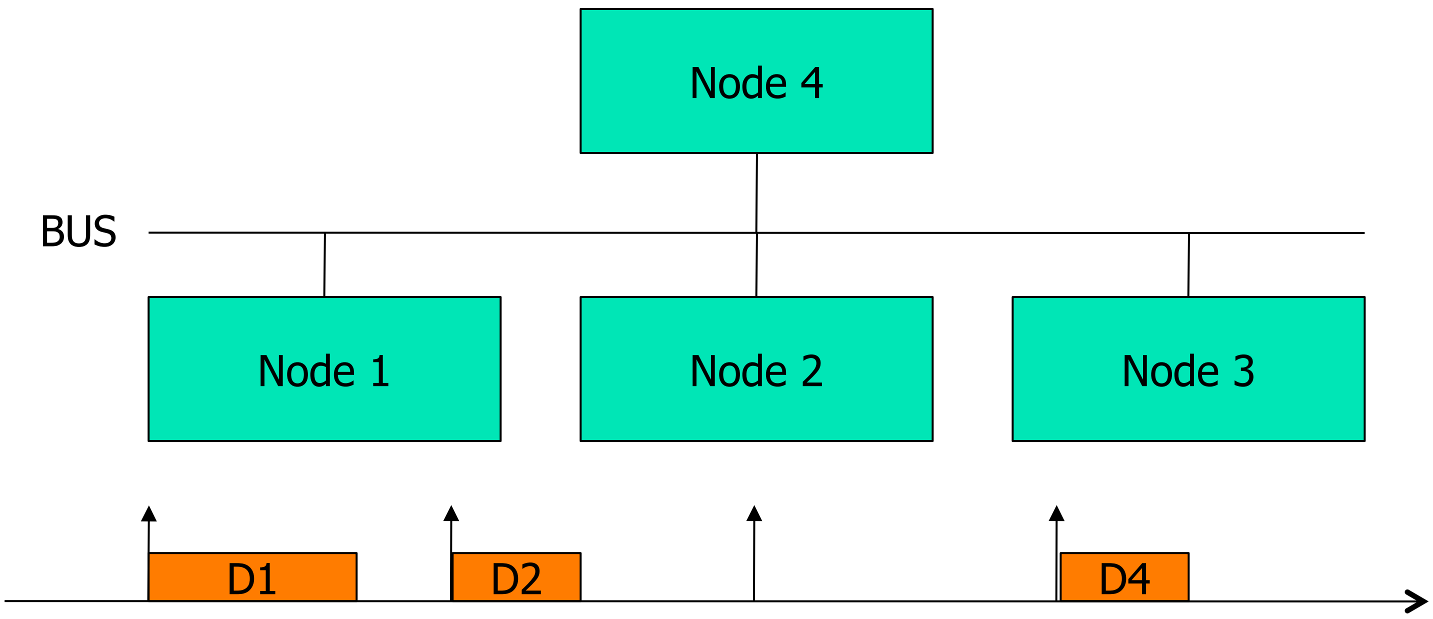


ProFiBus Bus

- A token circulates among certain nodes which behave like masters and send data to certain slaves
- Each master has counters and timers specifying the rotation time and the possession time of the token
- These parameters make it possible to detect the loss of the token or to force the release of the token.
- Not every node is necessarily the master.
- The token only circulates among a few masters to reduce turnaround times and token losses

MAC: Time Division

- Time-Division Multiple Access (TDMA)
- Time windows assigned to nodes
- Static periodic time table
- Synchronization by frame or global clock



Time Triggered Protocol

- Off-line static scheduling based on a table
- The nodes are identified by the issue date : there is no address field in the data
- Each site has exclusive "slots" on the bus, so the scheduling can be tested independently on each node

MAC: Collision Arbitration

- Carrier-Sense Multiple Access (CSMA)
  - No collision prevention
  - Re-transmission on collision
- CA: Collision Arbitration / Avoidance
  - A mechanism (priority) avoids the collision for a transmitter
  - The others retransmit later
- CD: Collision Detection
  - On collision, re-transmission for all after a random time determined independently for each

Controller Area Network (CAN)(1/2)

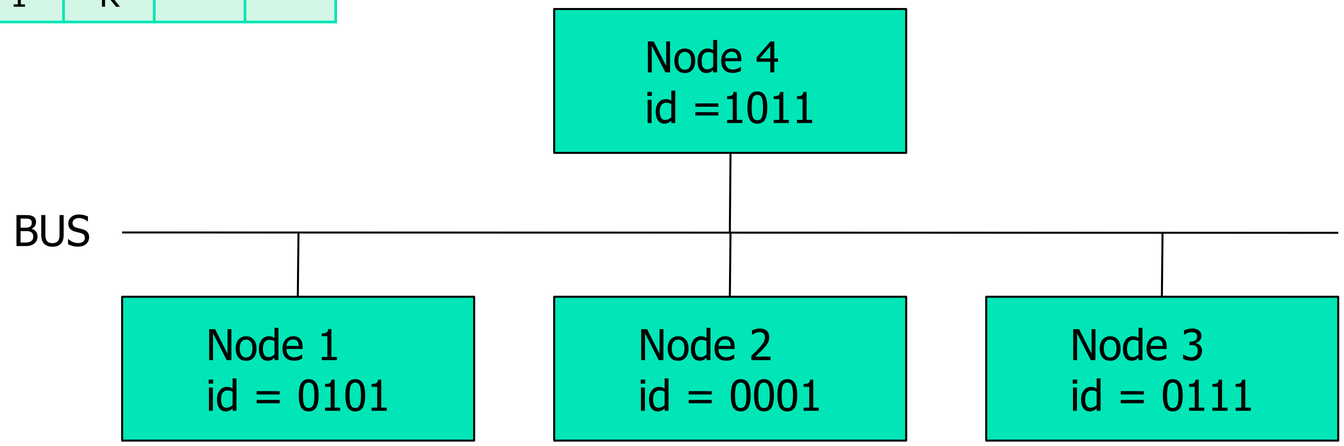
- Automotive field bus
- When the bus is free, the sender writes first its identifier bit by bit
- Each sender reads what is transmitted bit by bit
- In case of difference it becomes a receiver
- Priority is given according to the identifier
- The node with small identifier has the highest priority
- On rejection, it transmits again when the bus is free
- The computation of communication time is similar to the computation of a task response time with non-preemptive fixed priority scheduling

Controller Area Network (CAN)(2/2)

- In this example the CAN bus is said to be wired AND
- If a node sends a 0 bit, all nodes receive 0 bit
- If a node sends a (recessive) 1 bit, but reads a (dominant) 0 bit, it loses access to the bus and becomes a receiver

T →

N1	0	1	R	
N2	0	0	0	E
N3	0	1	R	
N4	1	R		



AFDX

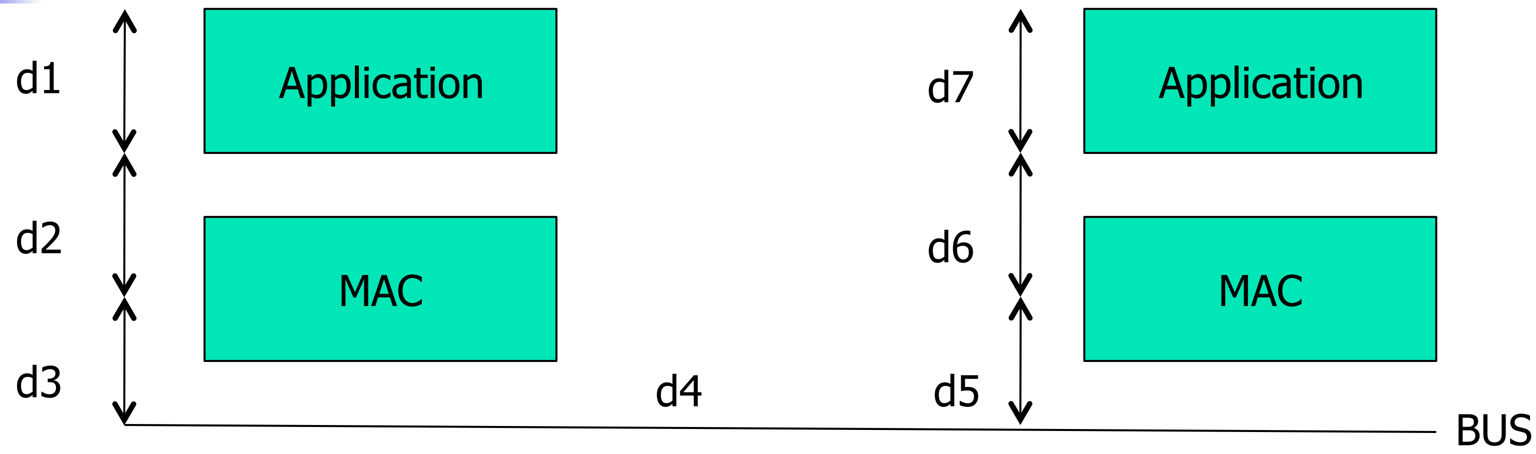
- Avionic Full DupleX switched Ethernet
  - Closer to an Ethernet network than a field bus
  - Complies with the ARINC 664 standard
- Physical Layer
  - Different from "open" Ethernet because no collision
  - 2 Rx / Tx twisted pairs between two connected nodes
  - Switches with buffers between nodes
  - Redundancy for safety and management of duplicates
  - Configuration of real links and switches to ensure virtual links of any 2 nodes
  - Multiplexing and frame management to ensure the co-habitation of several virtual links per node

Response Time

- Tasks
- Processor
- Capacity
- Response Time

- Messages
- Communication bus
- Time
  - To read/write buffers
  - To resolve collision
  - To transfer data
- Communication Time

Several latencies



- Times d1 and d7 spent in the application (buffer read/write)
- Times d2 and d6 spent to access medium (collision)
- Difficult to determine because depends on the access protocol
- Times d3, d4 and d5
- Easy to determine with data rate and size

Conclusions

- Solutions linked to equipment and technology (throughput, dependability, architecture, etc.)
- Solutions specific to the industrial sector
- Reduced network layers (close systems) compared to the Internet (open systems)
- Thanks to the analogy between tasks and messages, and processors and networks, (non-preemptive) scheduling theory allows to compute worse case response times
- The key element here is arbitration policy