ArtistDesign
Embedded Systems Seminar
Brussels, Belgium
June 18-19, 2009

Networks and Middleware

Luis Almeida
IEETA / DEEC-University of Porto, Portugal

Operating Systems and Networks Cluster
Real-Time Networks Activity

What is this about

- It's all about information!
  - **Information holding** — **Static** (time invariant)
  - **Information processing** — Transformation in the **value domain** (logic)
  - **Information transmission** — Transformation in the **space domain**

Dynamic (need time!)

Here we deal with

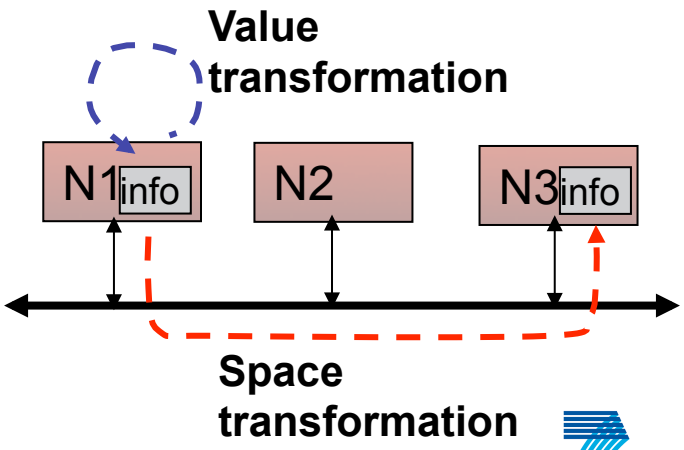
Information transmission

The operator for this transformation is

→ The network

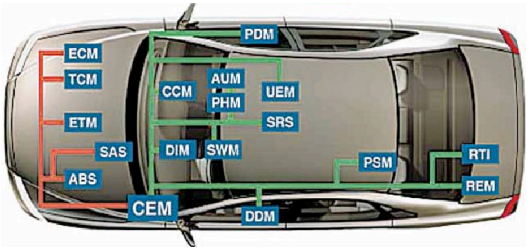
And for time-bounded transformations

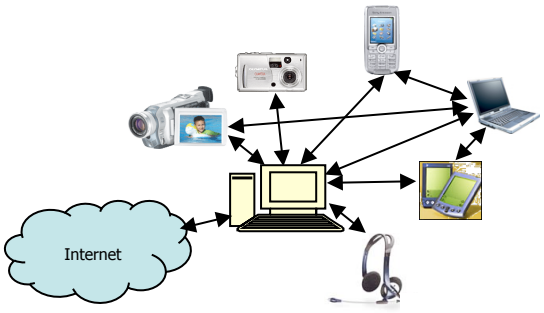
→ Real-Time Networks !



Many related networking frameworks

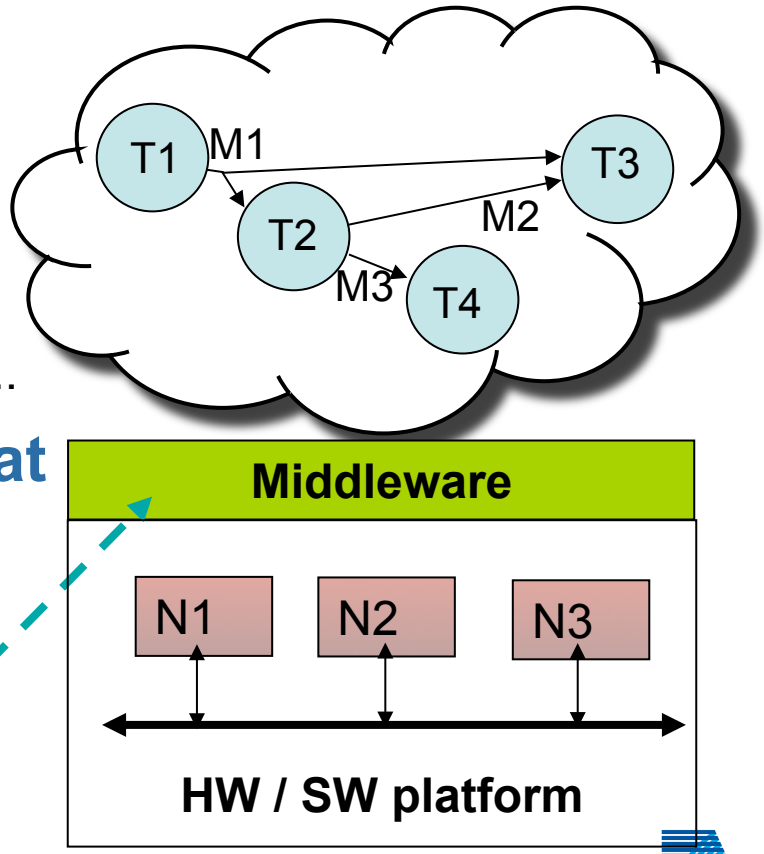
- Distributed embedded systems (DES)
- Networked embedded systems (NES)
- Ubiquitous systems
- Wireless sensor networks (WSN)
- Mobile ad-hoc networks (MANET)





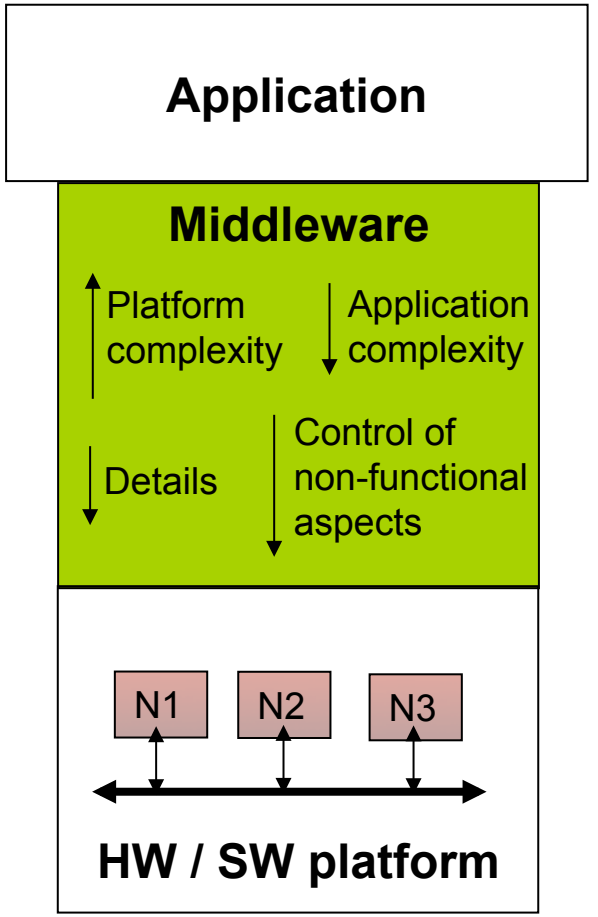
Abstracting away platform details

- **Applications are executed on HW / SW platforms**
  - Computing and communication
- **Implying many idiosyncrasies**
  - Dependence on HW / SW features
    - Processors, OSs, network protocols...
- **How to develop applications that**
  - Are agnostic to such idiosyncrasies?
  - Still delivering their services and exhibiting the desired properties...
  - And executing on a distributed platform...



Middleware: providing logical services

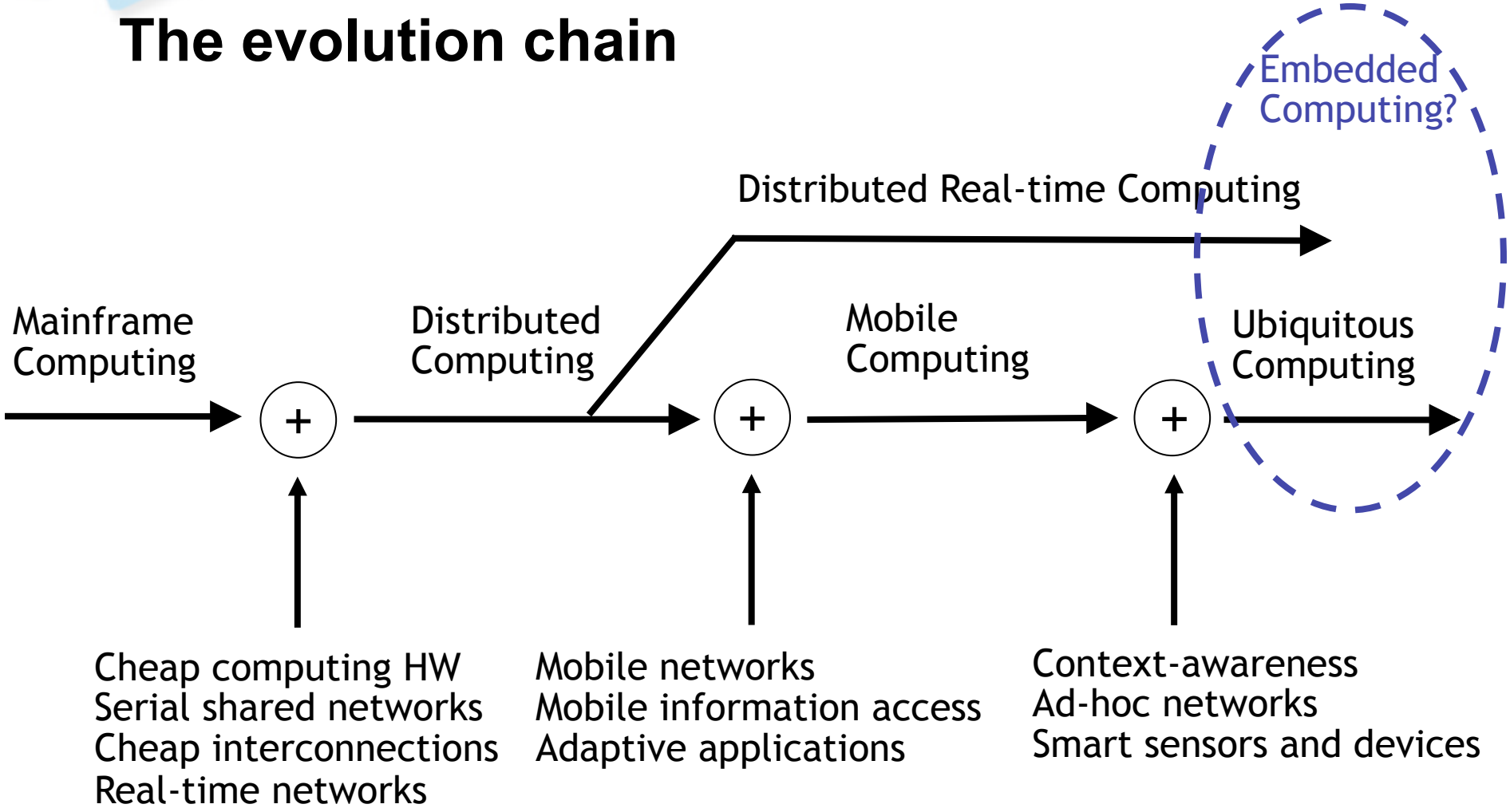
- **The middleware is a SW layer that**
  - Hides unnecessary details to the application (e.g., distribution)
  - Simplifies development, adds new services
- **But it implies trade-offs**
  - The HW and low-level SW have a profound impact on non-functional properties
    - timing, performance, dependability...
  - The simpler it is to develop applications the more complex the platform is
  - If the **right properties** are not properly enforced in the **lower layers** it will be hard (ineficient) to enforce them on the upper ones



In this talk...

- *Distributed* vs *ubiquitous / networked* ES
- Timing issues in the network
- Inside the protocol stack
  - The Physical Layer (PHY)
  - The Data Link Layer (DLL)
  - The Network Layer (NL)
  - The Application Layer (AL)
- Current technologies
- Some open issues

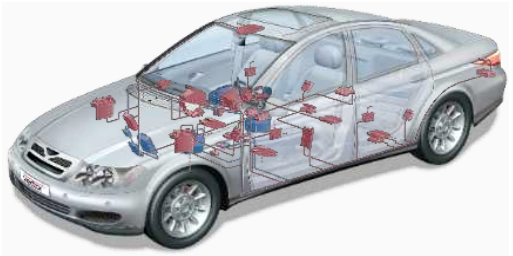
The evolution chain



Adapted from
(Strang and Linnhoff-Popien, 2004)

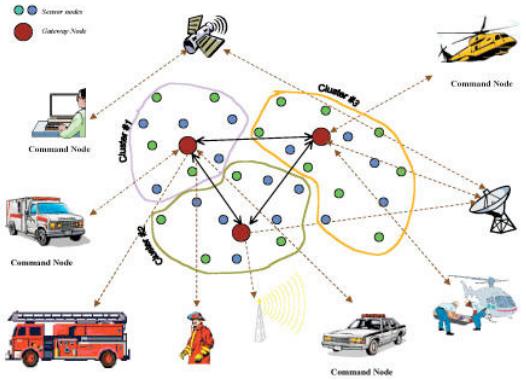
Distributed vs networked

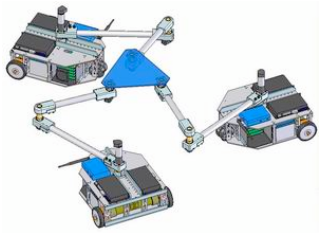
- **Distributed Embedded Systems**
  - System-centered (designed as a whole)
    - Confined in space (despite possibly large)
    - Normally fixed set of components
    - **Preference for wired networks**
      - **Fixed topology**
      - **Most interactions in single-hop segments**
  - Most frequent non-functional requirements
    - Real-time
      - End-to-end constraints on response to stimuli
      - Jitter constraints on periodic activities
    - Dependability
      - Ultra high reliability and safety, high availability
    - Composability
    - Maintainability



Distributed vs networked

- **Ubiquitous / networked Embedded Systems**
  - Interconnected stand-alone equipment for extra functionality (communication-centered)
    - Fuzzy notion of global system (and its frontiers)
      - too large / variable set of components
    - **Wireless/wired networks**
      - **Structured / Ad-hoc connections**
      - **Varying topology**
      - **Multi-hop communication**
  - Most common non-functional requirements
    - Scalability
    - Heterogeneity
    - Self-configuration
    - (Soft) real-time





Communication requirements

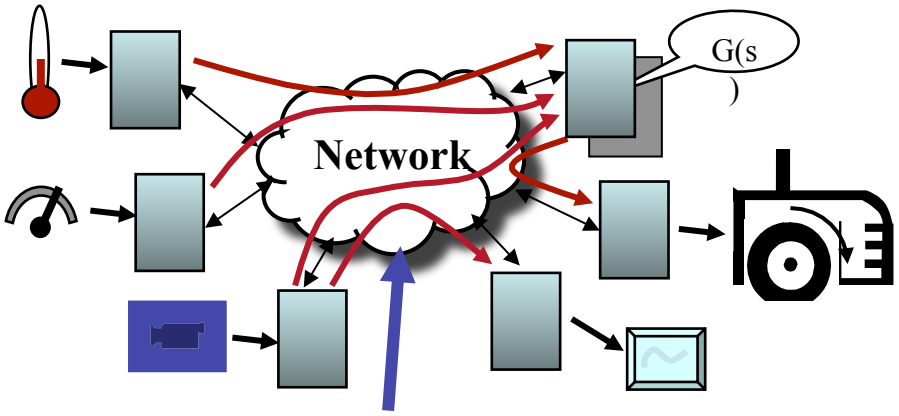
- **In both previous cases (DES + U/NES)**
  - Efficient transmission of **short data** (few bytes)
  - **Periodic** transmission (monitoring, feedback control) with **short periods** (ms) and **low jitter**
  - **Fast transmission** (ms) of **aperiodic requests** (alarms)
  - Transmission of **non-real-time data** (configuration, logs)
  - **Multicasting**

State messages

Event messages

Real-time messages

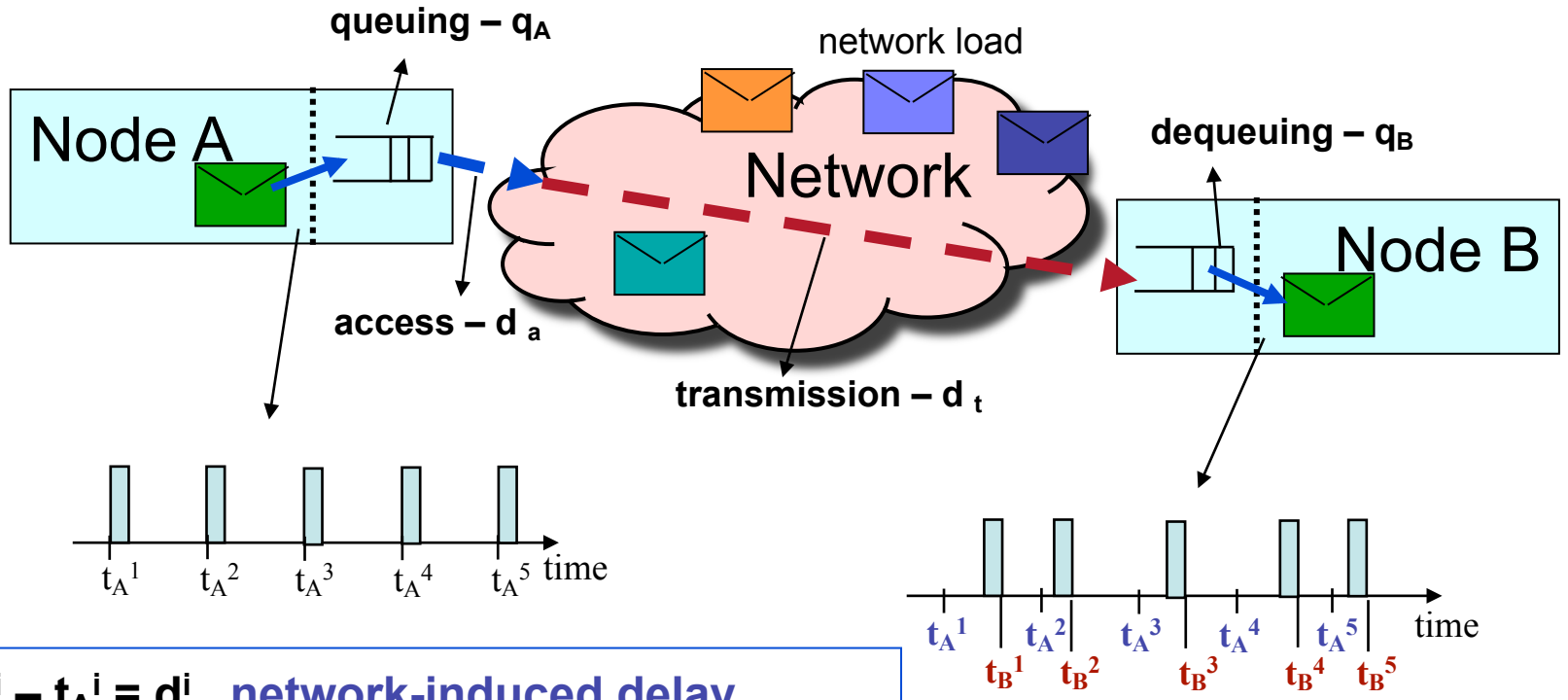
! There is also a growing interest on **real-time multimedia traffic** (e.g., machine vision)
➔ **long** RT data



Bandwidth:
Limited shared resource

Timing Issues in the Network

Network delay and delay jitter



$t_B^i - t_A^i = d^i$, **network-induced delay**

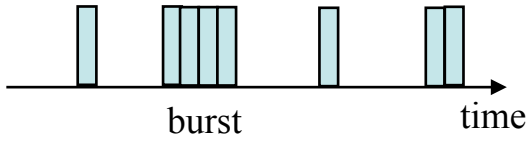
$d^i = q_A^i + d_a^i + d_t^i + q_B^i$, **delay components**

$d^i - d^{i-1} = j^i$, **delay jitter**

Reception instants may suffer irregular delays due to interferences from the network load, queuing policies and processor load

Burstiness and throughput

- **Burstiness – measure of the load submitted to the network in a short interval of time.**
  - Bursts have a profound impact on the real-time performance of the network and impose high buffering requirements.
    File transfers are a frequent cause of bursts.

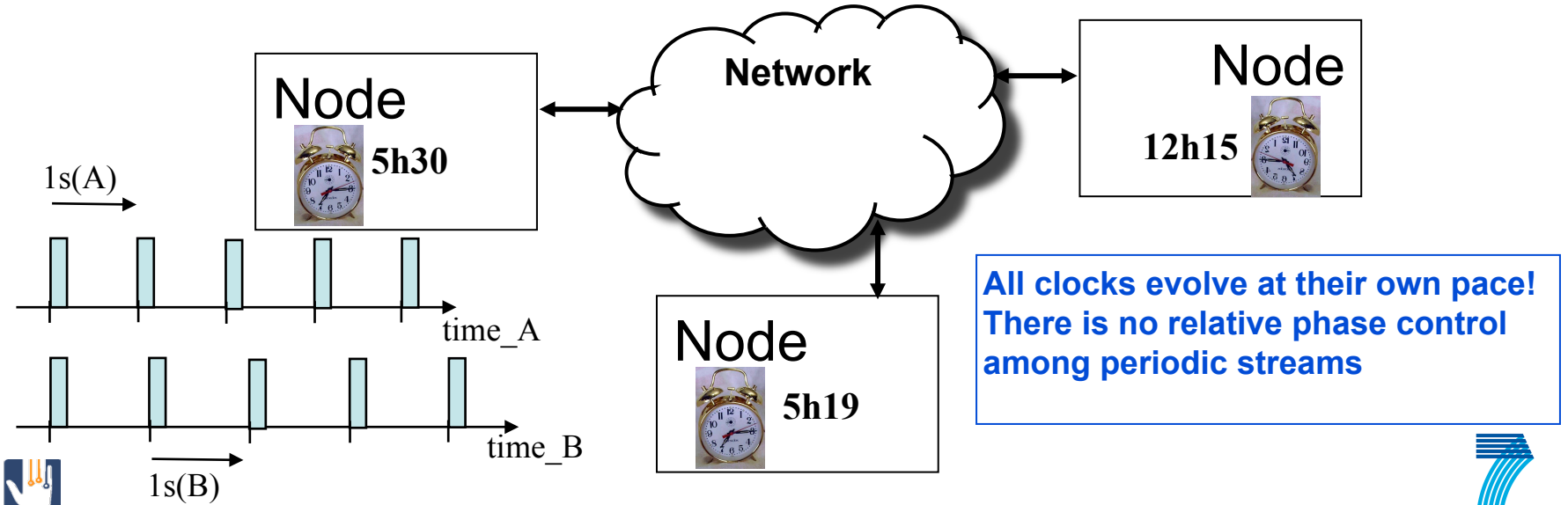


- **Throughput – average amount of data, or packets, that the network dispatches per unit of time (bit/s and packets/s).**
- **Arrival / departure rate – long-term rate at which data arrives at/ from the network (bit/s and packets/s).**
- **Capacity – maximum (gross) bit-rate ach… …e network**

Network bandwidth

Time across a network

- **As opposed to a centralized system, in a distributed system each node has its own clock**
  - Without specific support, there is **no explicit coherent notion of time** across a distributed systems
  - Worse, due to **drift**, clocks tend to permanently diverge



Time across a network

- **However, a coherent notion of time can be very important for several applications to:**
  - Carry out **actions** at **desired time** instants
    - e.g. synchronous data acquisition, synchronous actuation
  - **Time-stamp** data and events
    - e.g. establish causal relationships that led to a system failure
  - Compute the **age** of data
  - **Coordinate** transmissions
    - e.g. TDMA clock-based systems

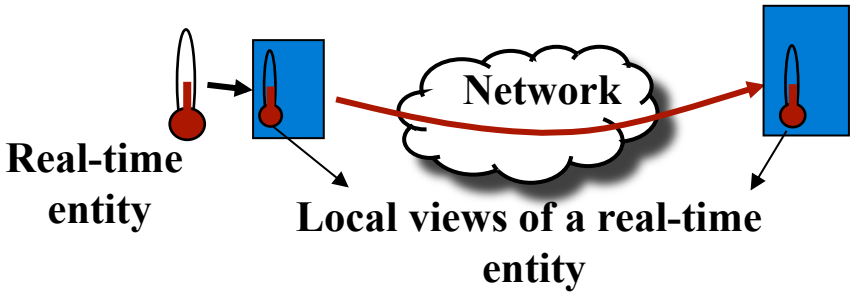
But how to synchronize the clocks across the network?

Synchronizing clocks

- **Clocks can be synchronized:**
  - **Externally** – an external source sends a time update regularly (e.g. GPS)
  - **Internally** – nodes exchange messages to come up with a global clock
    - **Master-Slave** – The time master spreads its own clock to all other nodes
    - **Distributed** – All nodes perform a similar role and agree on a common clock, for example, using an average (e.g. FTA, Fault-Tolerant Average)
- Standards: **NTP, SNTP, IEEE 1588**
- **Uncertainties in network-induced delay** lead to limitations in the achievable **precision**
  - Typical precision with SW methods in small networks is worse than 10μs
  - With special HW support, it is possible to reach 1μs or better

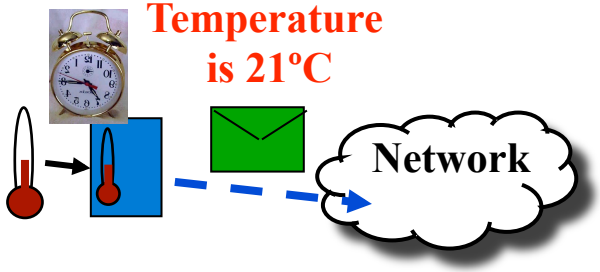
Constraints on the network delay

- **Real-time messages must be transmitted within precise time-bounds**
  - to assure **coherence** between senders and receivers concerning their **local views** of the respective real-time entities

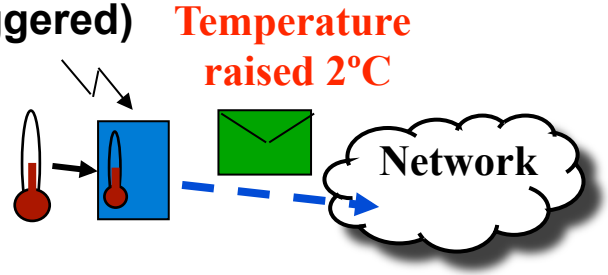


- This applies to both **event and state messages**

State (time-triggered)

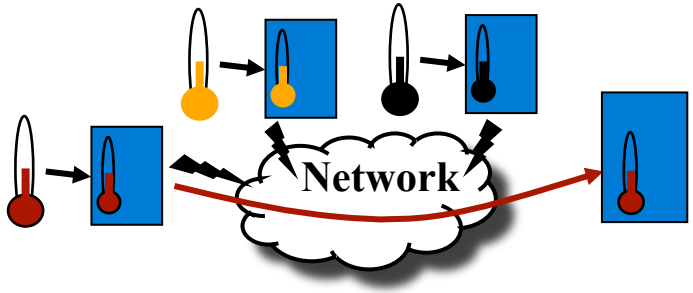


Event / State change (event-triggered)



Event and time-triggered messages

Event-triggered



transmitter receiver

Temperature raised 2ºC — t_1 m_1

Temperature decreased 2ºC — t_2 m_2

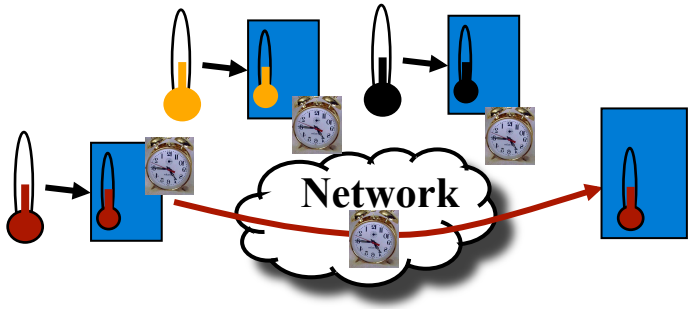
time

What if: m_2 is lost?

Notice that: $\Delta t=t_2-t_1$ is unbounded

Typical solutions involve limiting Δt (e.g. with heartbeat)

Time-triggered



transmitter receiver

Temp = 20ºC t_1 m_1

Temp = 20ºC t_2 m_2

Temp = 21ºC t_3 m_3

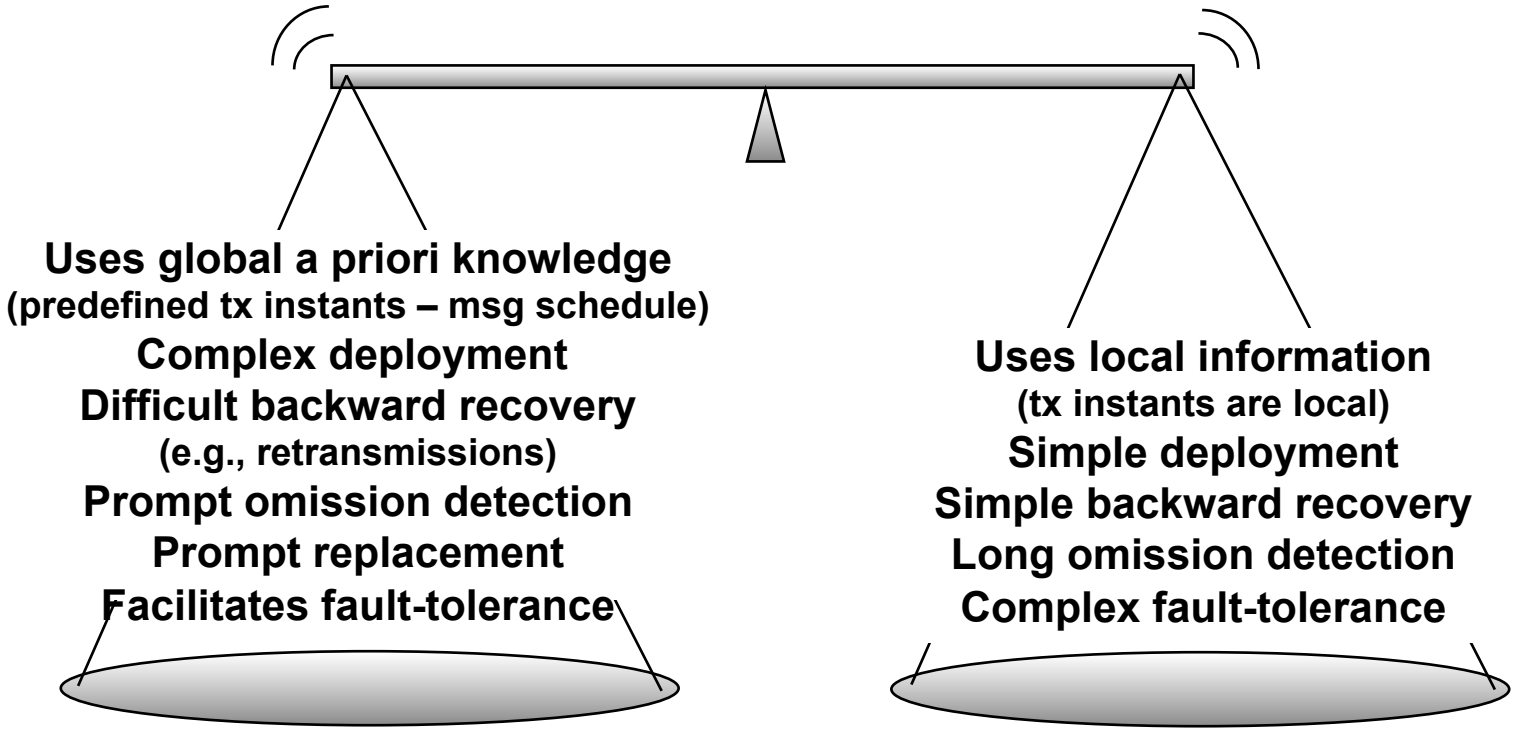
Temp = 22ºC t_4 m_4

time

Notice: $\Delta t=t_i-t_{i-1}$ set according to system dynamics

Losing one message causes inconsistency during Δt

Event vs time triggering

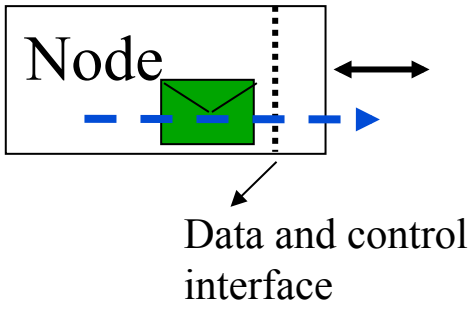


Time-triggered network **Event-triggered network**

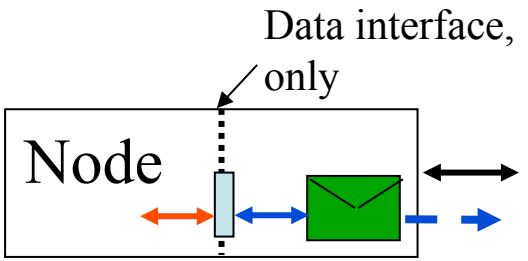
How to support event & state messages efficiently?

Transmission control

- **Determines who triggers network transactions, application or network**
  - **External control**
    - Transactions are **triggered** upon explicit control signal from the **application**.
    - Messages are queued at the interface.
    - **Highly sensitive to application design/faults.**

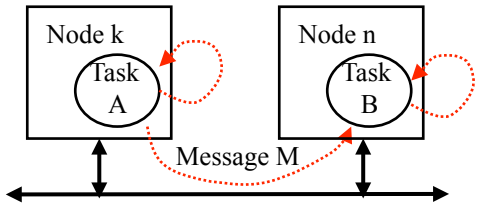


  - **Autonomous control**
    - The network **triggers** transactions **autonomously**.
    - No control signal crosses the CNI.
    - Applications **exchange data** with the network by means of **buffers**.
    - **Deterministic behavior.**

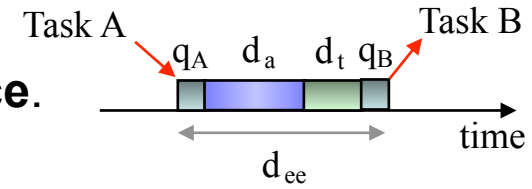


Information flow

- **Determining end-to-end delay**

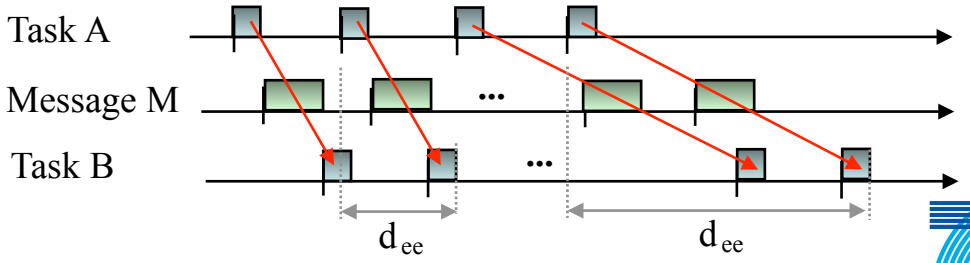


  - **ET-network with external control**
    - Transactions are composed of several **elementary actions** carried out **in sequence**.



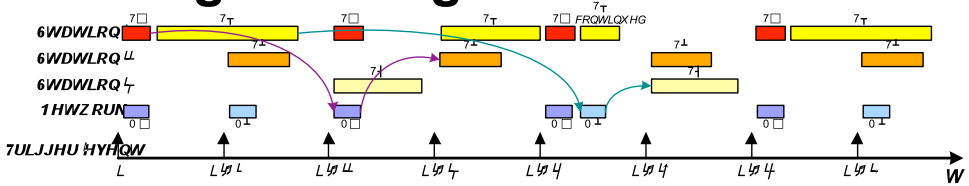
  - **TT-network with autonomous control**
    - The **elementary actions** in each intervenient (transmitter, network, receiver) are **decoupled**, spinning at an appropriate rate.

Requires relative phase control to avoid high delays and jitter

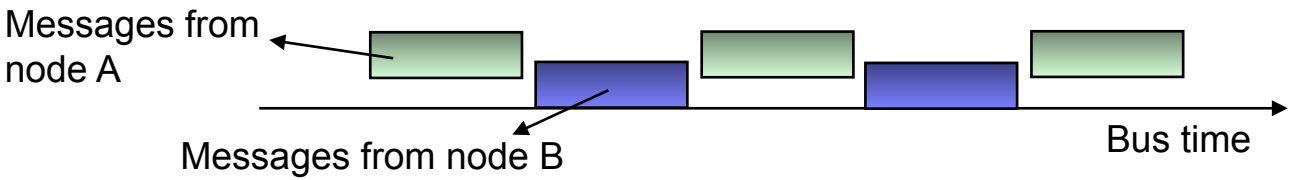


Information flow

- **In a TT-network**
  - The **tight control** of the **relative phase** required between all system activities (message transmissions and task executions) **imposes rigid architectural constraints**
    - **Time-triggered architecture**
  - The **whole system** must be **designed altogether** (network and nodes)



  - However, once the network is designed, **nodes will not interfere with each other** (transmissions occur in disjoint intervals)
    - **Composability with respect to temporal behavior**



Inside the protocol stack

Physical entities & technologies

- **Computing - nodes**
  - Desktop and Laptop PCs
  - Embedded PCs
  - Single-board computers
  - Micro-controllers, specific processors, FPGAs, ...

DES

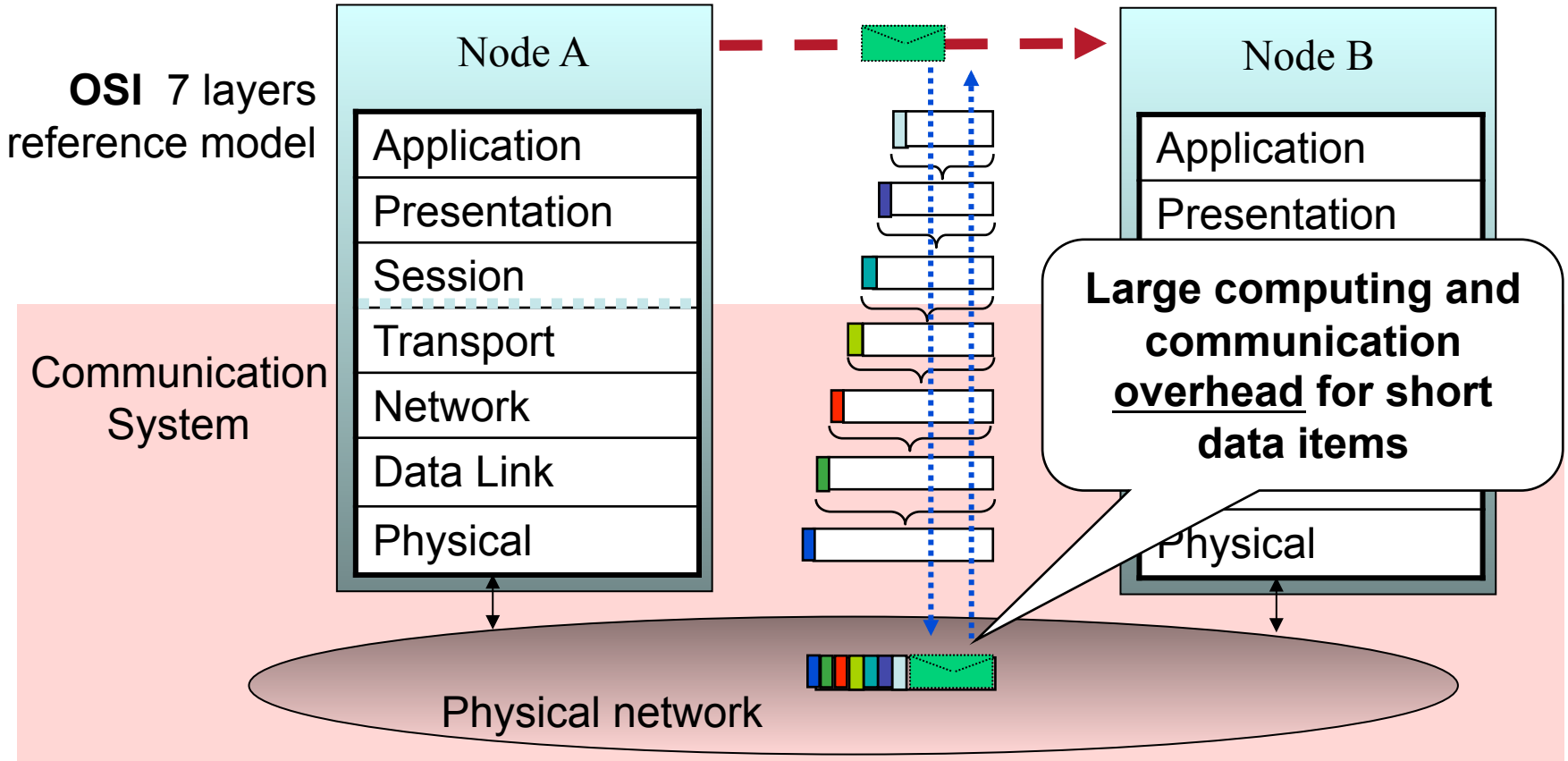
- **Communication - networks**
  - Public telephony networks (ADSL, GPRS, UMTS...)
  - Local Area Networks (Ethernet, WiFi, ...)
  - Control networks (CAN, FlexRay, PROFIBUS, ...)

DES

  - Powerline networks (X10, HomePlug, ...)
  - Personal Area Networks (Bluetooth, IrDA, ...)
  - Sensor Networks (ZigBee, RFID, ...)

! Several of these technologies were developed for general purpose systems → potential impact on efficiency...

The *not only physical* communication process
(the OSI protocol stack)

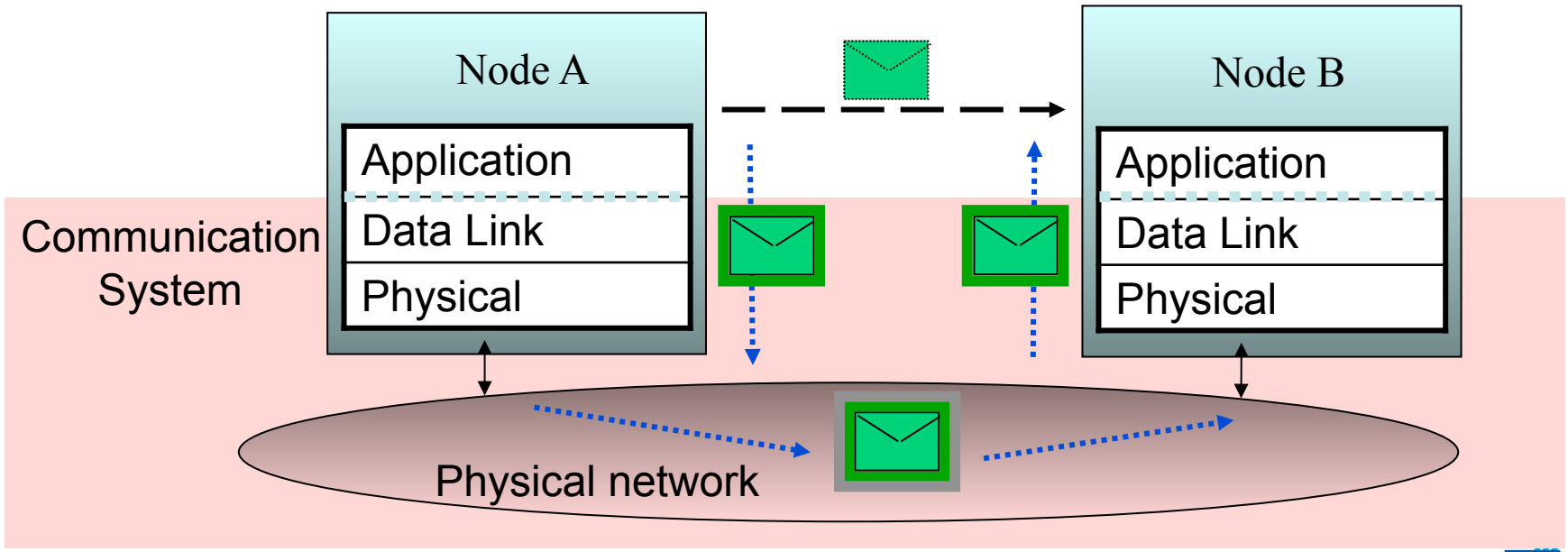


Requirements for an embedded / real-time protocol stack

- **The end-to-end communication delay must be bounded**
  - All services at all layers must be **time-bounded**
  - Requires appropriate **time-bounded protocols**
- **The 7 layers impose a considerable overhead…**
  - **Time to execute** the protocol stack (can be dominant wrt tx time)
  - **Time to transmit** protocol control information (wrt to original data)
  - **Memory requirements** (for all intermediate protocol invocations)
- **Many embedded / real-time networks**
  - are dedicated to a well defined application (no need for presentation)
  - use single broadcast domain (no need for routing)
  - use short messages (no need to fragment/reassemble)

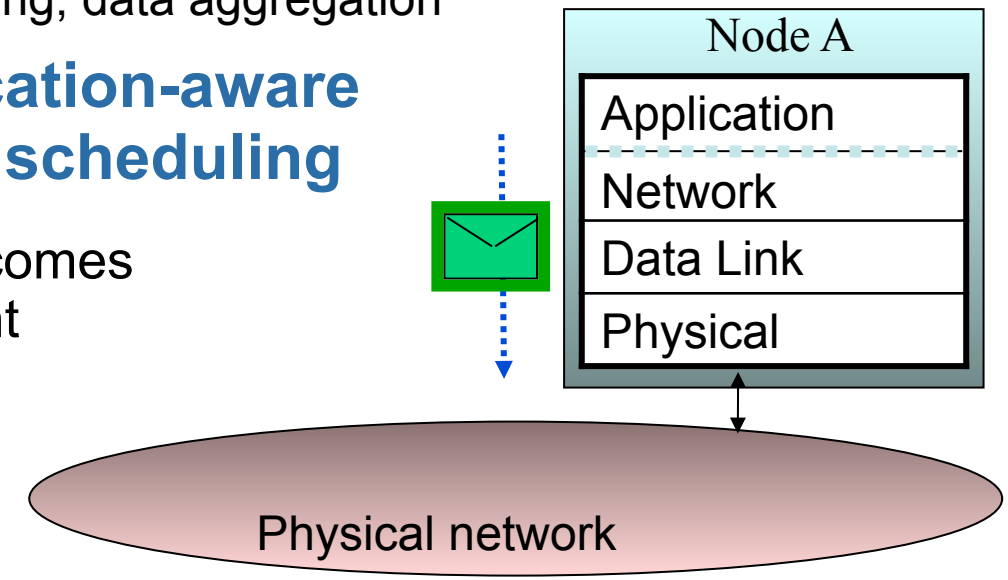
OSI collapsed model

- **Application services access the Data Link directly**
  - Services from other layers maybe present
- **In process control and factory automation these networks are called Fieldbuses**



Ubiquitous systems (e.g., WSN)

- **A network layer with routing services is fundamental**
  - multi-hop communication, logical addresses
- **Some more complex application layers**
  - Specific middlewares with certain services
    - Localization, tracking, data aggregation
- **Energy-aware / location-aware routing and traffic scheduling**
  - Synchronization becomes particularly important



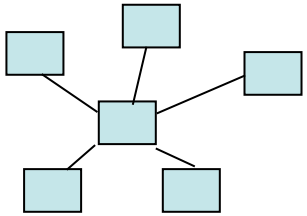
Issues in the Physical Layer

Physical layer

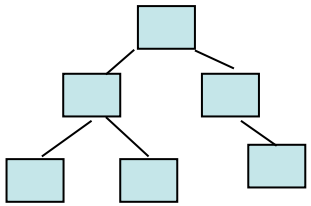
- **Issues related with the physical layer:**
  - **Interconnection topology**
  - **Physical medium**
  - Coding of digital information
  - **Transmission rate**
  - Maximum interconnection length
  - Max number of nodes
  - Feeding power through the network
  - **Energy Consumption**
  - Immunity to EMI
  - Intrinsic safety

Physical layer

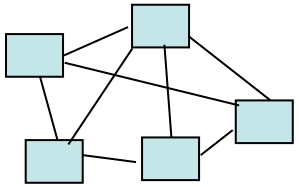
- **Interconnection topology**



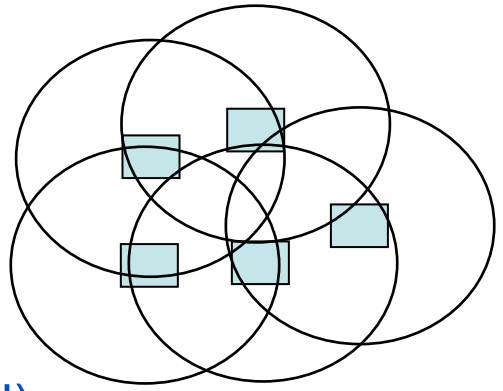
Star



Tree

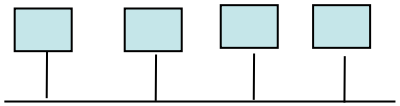


Mesh (wired)

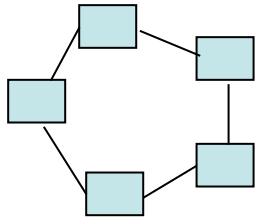


Mesh (wireless-RF)

- **Physical medium**
  - **Copper wiring**
  - Optical fibers
  - **Wireless – Radio Frequency**
  - Wireless – Infra-red light



Bus



Ring

Physical layer

- **Propagation delay (δ = L/V)**
  - Caused by the **limited speed** of the electromagnetic wave
    - Typically, about **5ns/m in copper cables**, **3.3ns/m in the air**
- **Bit length (b = δ*Tx_rate)**
  - Number of bits traveling on the medium at the same time
    - High bandwidth networks always have b>1
    - b<1 → all nodes “see” the same bit at a time (CAN)
- **Wireless propagation**
  - **Strong attenuation**
    - **Free-space attenuation model**, increases with log of distance
    - Establishes a **communication range** and an **interference range**
    - Cause of **hidden-nodes** but also allows **spatial channel reuse**
  - **Multi-path fading, directional antennas, asymmetric links...**

Physical layer

- **Energy consumption**
  - Very **complex relationship** among transmission power, bit rate, frame rate, network traffic, error control coding, retransmissions policy, type of medium access control and physical bit encoding. **Need to trade-off**.
- **Example of energy trade-offs in wireless**
  - **Higher data rate** uses more energy but **transmissions take less time** (however, probability of errors and retransmissions also increases)
  - In a mesh-like network (WSNs), **higher tx power** might **reduce the number of hops**, thus less energy spent in storing and forwarding
  - **Asynchronous transmissions** (ET-like) reduce power on tx but **receivers may need to be awake** all the time!
  - **Collisions require retransmissions** but **avoiding** them requires a **synchronization mechanism**

Issues in the Data Link Layer

Data link layer

- **Issues related with the data link layer:**
  - **Addressing**
  - **Logical link control – LLC**
    - Flow control
    - Transmission error control
  - **Medium access control – MAC**
    (for shared media)

Data link layer

- **Addressing**
  Identification of the parts involved in a network transaction
  - **Direct addressing**
    The **sender and receiver(s) are explicitly identified** in every transaction, using physical addresses (as in Ethernet)
  - **Indirect (source) addressing**
    The **message contents** are explicitly identified (e.g. temperature of sensor X). Receivers that need the message, retrieve it from the network (as in CAN and WorldFIP)
  - **Indirect (time-based) addressing**
    The message is **identified by the time** instant at which it is transmitted (as in TTP)

Data link layer

- **Logical Link Control**
  Deals with the transfer of network packets
  - Might have an **impact on the network delay** depending on whether **sender/receiver synchronization** is used
    - e.g., ack. mechanisms, retry mechanisms, request data on reply
    - **Retries mechanisms** (e.g., PAR, ARQ) might have a strong **negative impact** on the data **timeliness**...
      - No point on continue trying to transmit out-dated values

Data Link Layer

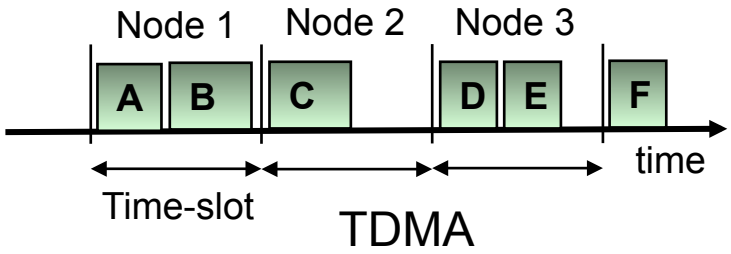
- **Medium Access Control**
  Determines the waiting time to access the network
  - It has a **large impact** on the **network delay**

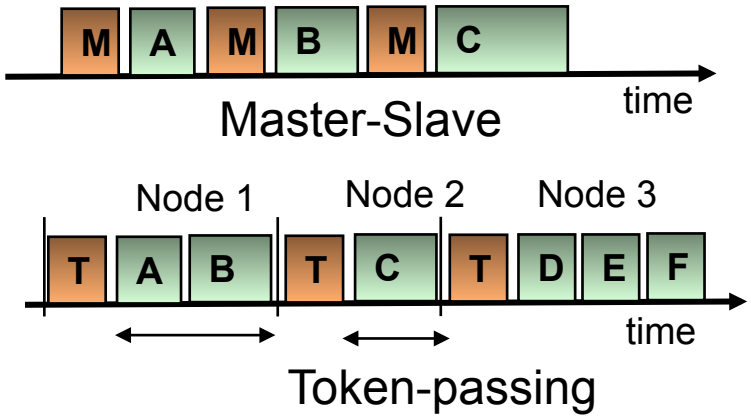
- **Two main families**
  - **Controlled access**
    - based on transmission control
      - The network says when to transmit
  - **Uncontrolled access**
    - based on arbitration (typically CSMA-based)
      - The nodes try to transmit immediately when needed

Temporal multiplexing

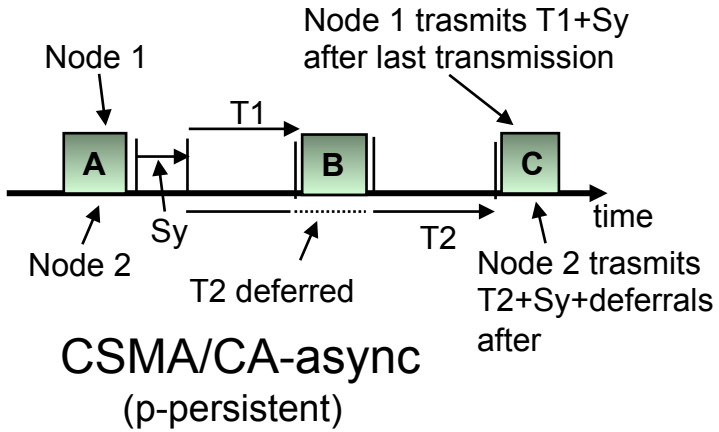
Data Link Layer

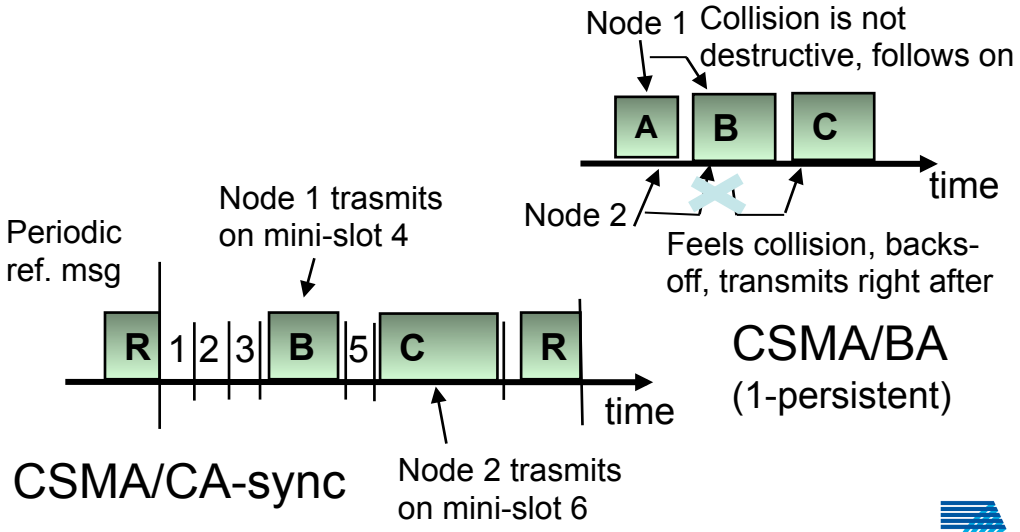
- **Controlled access**





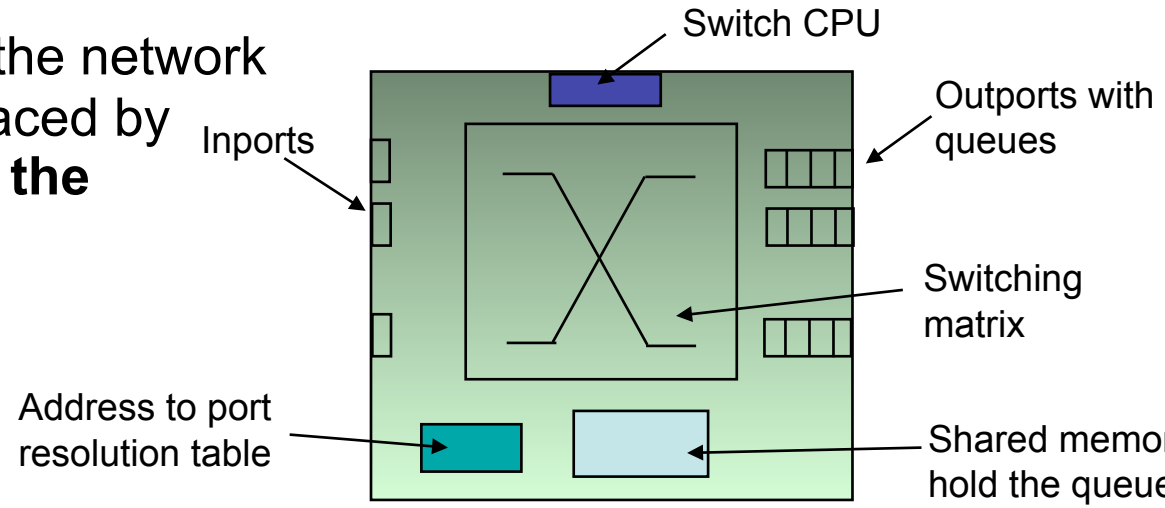
- **Uncontrolled access**





Data Link Layer

- **Frequency multiplexing**
  - Wireless channels (not truly isolated...)
- **Switched**
  micro-segmented network with central switching hub
  - Nodes send asynchronously messages to the switch using point-to-point links – **no shared medium, no collisions**
  - Contention at the network access is replaced by **contention at the output ports**



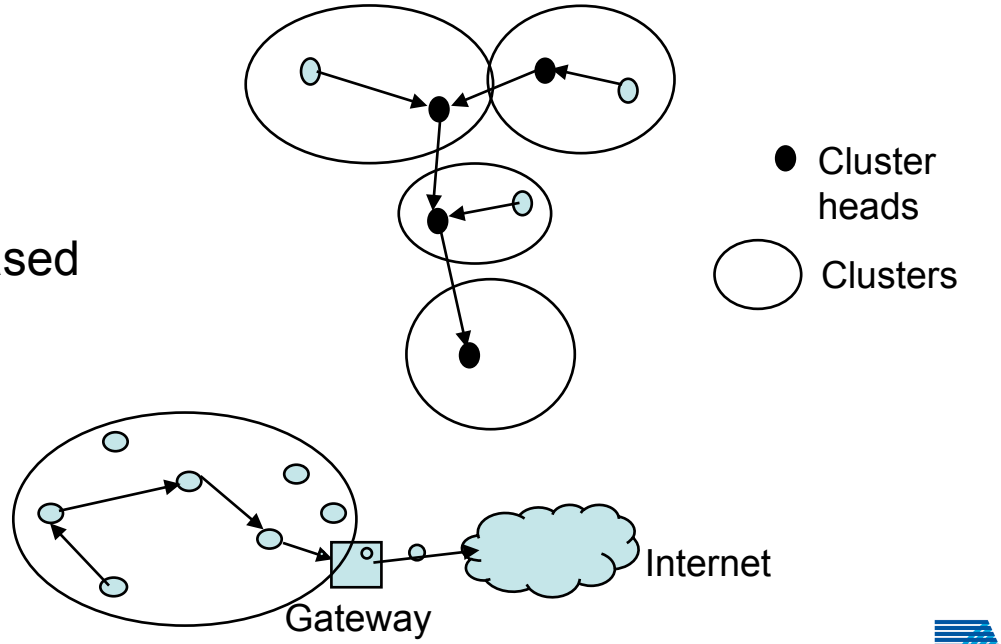
Issues in the Network Layer

Network layer

- **Issues related with the network layer:**
  - Logical addressing
  - Routing

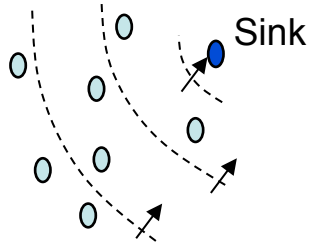
Network layer

- **Logical Addressing**
  - Independence of higher protocols wrt the physical hardware
  - Requires an ARP (address resolution protocol)
- **Routing**
  - **Ad-hoc routing**
    - Flat routing
    - Hierarchical / Cluster-based
  - **Inter-network routing**
  - **Setting routes**
    - Proactive
    - Reactive



Network layer

- **Channel reuse in space**
  - Nodes that are sufficiently apart can transmit simultaneously
    - communication range versus interfering range
- **Synchronized frameworks**
  - Communication occurs in slots that are defined across the network (TDMA fashion)
    - Require synchronization (clock, beacons)
  - Enables reducing the data routing delay by defining adequate transmission (slot) sequences
  - Enables controlling on-off switching for energy saving



Issues in the Application Layer

Application layer

- **Issues related with the application layer**
  - Set of services that are common for a class of applications
  - (Middleware issues)
    - Transparency wrt Distribution, OS, Languages...
    - Support for different programming paradigms (SO, OO, CB...)
  - **Cooperation model**
    - Client-Server
    - Producer-Consumer
    - Producer-Distributor-Consumer
    - Publisher-Subscriber

Application layer

- **Client-Server**
  - **Servers hold** information. **Clients request** it
    - Transactions **triggered by the receivers** (clients)
    - Typically based on **unicast** transmission (one to one comm.)
  - Transactions can be **synchronous** (client blocks until server answers) or **asynchronous** (client follows execution after issuing the request)
- **Producer-Consumer**
  - **Producers disseminate** information. **Consumers use** it
    - Transactions **triggered by the senders** (producers).
    - Based on **broadcast** transmission (each message is received by all)

Application layer

- **Producer-Distributor-Consumer**
  - Similar to Producer-Consumer
  - Adds a **central scheduler** to control Producers transmissions
- **Publisher-Subscriber**
  - Uses concept of **group communication**
  - Nodes must adhere to groups either as **publisher** (produces information) or as **subscriber** (consumes information)
  - Transactions are **triggered by the publisher** of a group and disseminated among the **respective subscribers**, only (multicast)

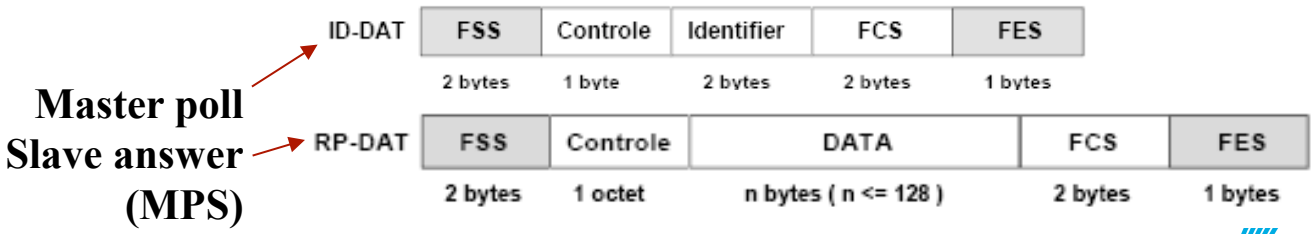
Communication technologies
for (Distributed / networks) embedded systems

WorldFIP

Factory Instrumentation Protocol

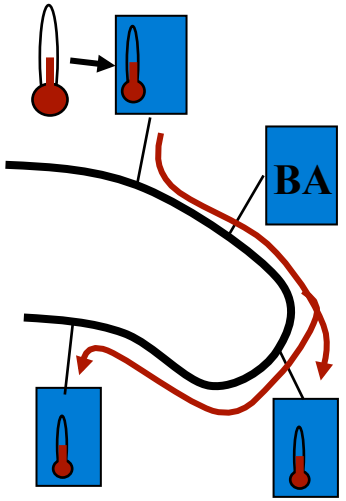
www.worldfip.org

- **Created, in the 80s, in France for use in process control and factory automation.**
- **2 messaging systems:**
  - **MPS – real-time** services, periodic, aperiodic;
    MMS (ISO 9506) **subset** – non-real-time messaging
- **Data payload between 0 and 128 bytes (256 for NRT)**
- Source-addressing **(message identifiers with 16 bits)**
- Master-Slave **bus access control**
  - BA - Bus Arbitrator



WorldFIP

- MPS – ***Messagerie Periodique e Sporadique***
- Producer-Distributor-Consumer **middleware**
- **Concept of** Network Variable
  - Entity that is distributed (several local copies coexist in different nodes)
  - Can be **periodic** or **aperiodic**
  - Local copies of **periodic variables** are **automatically refreshed** by the network
  - Local copies of **aperiodic variables** are refreshed by the network upon **explicit request**

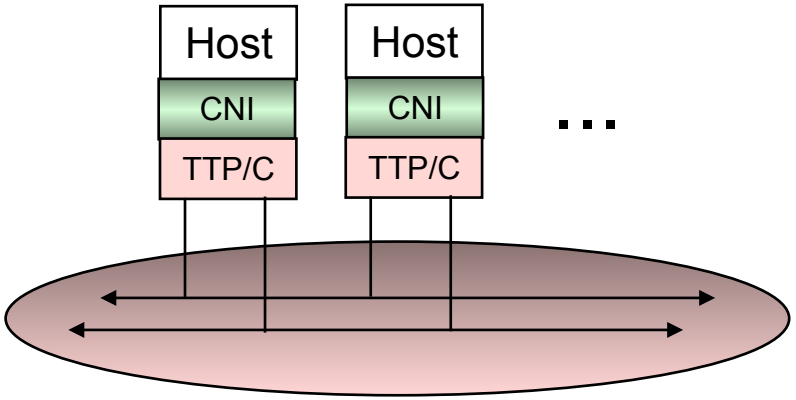


TTP/C

Time-Triggered Protocol

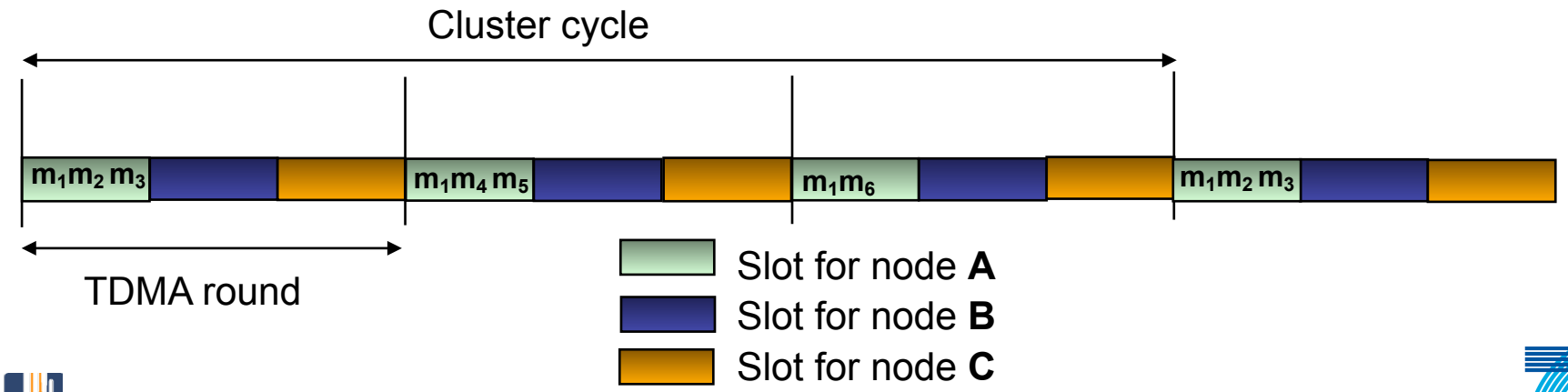
www.tttech.com

- Created around 1990 within the **MARS project** in the Technical University of Vienna
- Aims at **safety-critical** applications
- Considers an architecture with **nodes integrated in fault-tolerant units** (FTUs), interconnected by a **replicated bus**
- Includes support for **prompt error detection** and **consistency checks** as well as **membership** and **clock synchronization** services.



TTP/C

- **TDMA access** with **one slot** allocated **per node** and **per round**
- The periodic sequence of slots is a **TDMA round** (typical values 1-10ms)
- In **each slot** the respective node may send **one frame** (up to 240B)
- Each **frame** may contain **several messages**
- The periodic sequence of messages is a **Cluster cycle**
- All **message transmission** instants are stored in a **distributed static table**, the MEDL



PROFIBUS

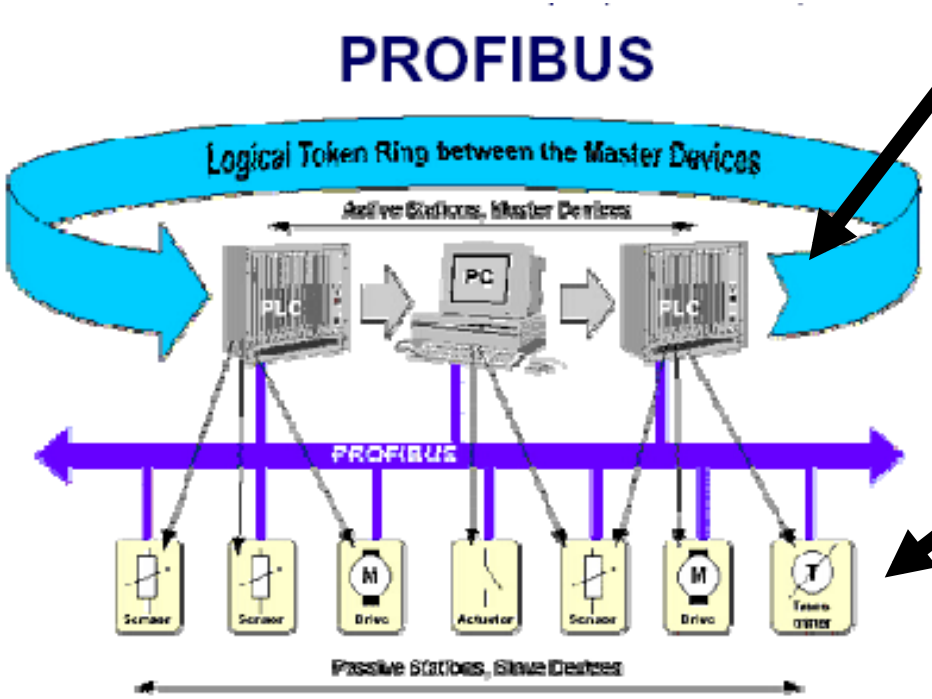
PROcess FieldBUS

www.profibus.com

- **Created in the late 80's by Siemens, in Germany**
- **Two main application profiles (client-server middlewares):**
  - **PROFIBUS / FMS** - Fieldbus Message Specification
  - **PROFIBUS / DP** - Decentralised Peripherals
    - The most common profile (~90%)
- **Data payload between 0 and 246 bytes**
- Direct-addressing **(1 byte, possibly extended)**
- Hybrid **bus access control**
  - **Token-passing** among masters
  - **Master-Slave** in each individual data transaction

PROFIBUS

- General architecture



Masters:
controllers,
regulators, …
(PC, PLC,…)

Slaves:
sensors,
actuators
(motors, valves,
thermometers…)

CAN

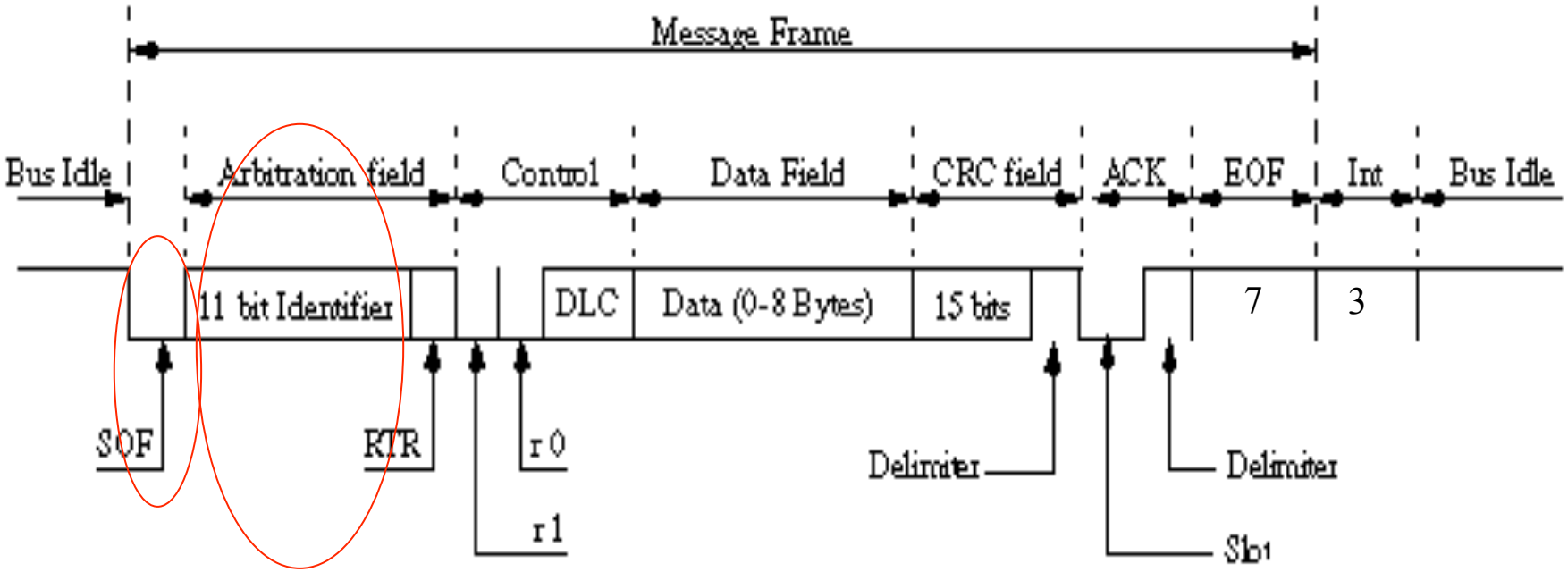
Controller Area Network

www.can-cia.de

- Created in the early 90s by Bosch, GmbH, for use in the automotive industry.
- Data **payload** between **0 and 8 bytes**
- **Source-addressing** (message identifiers) with 11 bits in version A and 29 bits in version B
- **Asynchronous bus access** (CSMA type)
- **Non-destructive arbitration** based on message identifiers (establish priority)
  - Bit-wise deterministic collision resolution CSMA/BA (CA?,DCR?)

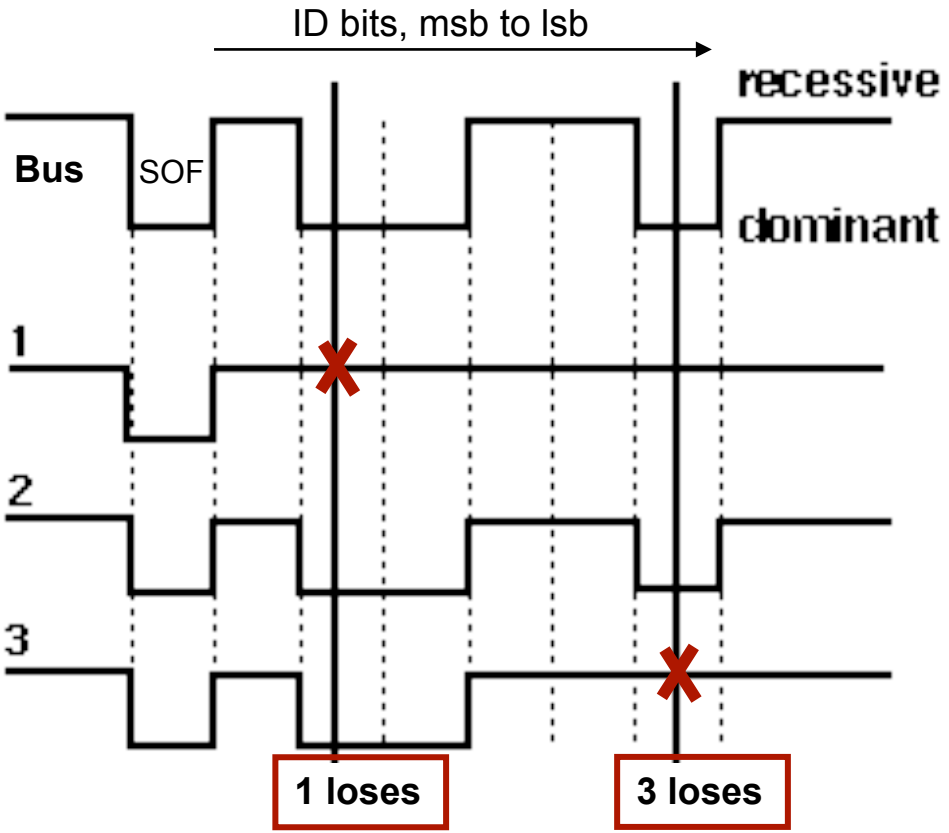
CAN

- **CAN 2.0A message frame**



CAN

- **Prioritized arbitration**
  - All nodes transmit and listen every bit
  - If different, then lost arbitration and backoff



CANopen – a middleware for CAN

CAN-in-Automation

http://www.can-cia.com

- **Key features:**
  - **Process data** shared with the **producer/consumer** model
  - Standardized **device description** and **methods**
    - data, parameters, functions, programs
  - Standardized services for **device monitoring**
    - e.g. membership functions based on heartbeats
  - **System services**: synchronization message, central time-stamp message (e.g. synchronous data acquisition)
  - **Emergency messages**
  - Adopted by other protocols (e.g. Ethernet POWERLINK)

CANopen

- **Process Data Objects (PDO)**
  - Carry actual application data; broadcast, producer/consumer cooperation model, unacknowledged
    - Asynchronous PDOs (event-triggered)
    - Synchronous PDOs (time-triggered based on Sync Object)
- **Service data objects (SDO) -** device configuration
  - Read/write device OD entries, following sync client/server model
- **Network management (NMT)**
  - Node monitoring
  - Control their communication state

Ethernet

standards.ieee.org/getieee802/802.3.html

- Created in the mid 70s (!) at the Xerox Palo Alto Research Center.
- CSMA/CD non-deterministic arbitration **(outdated...)**
  - 1-persistent transmission (transmits with 100% probability as soon as the medium is considered free)
  - Collisions can occur during the interval of one slot after start of transmission (512 bits)
  - When a collision is detected a jamming signal is sent (32 bits long)
- Frames vary between 64 (min) and 1518 (max) octets

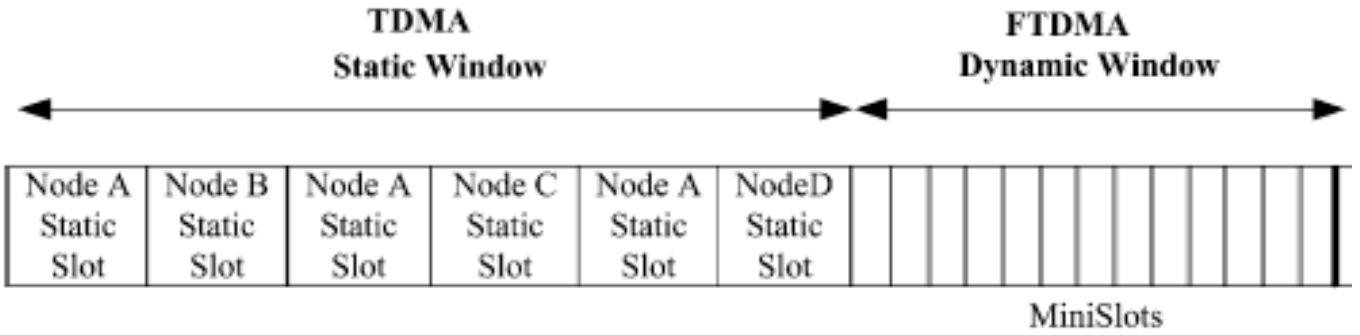
Ethernet

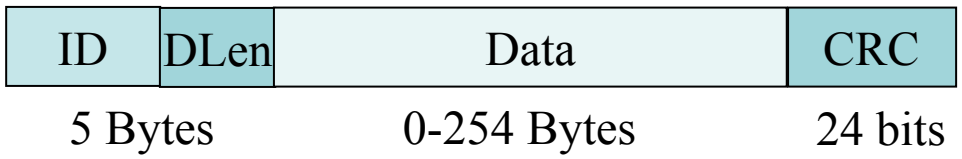
- **Using switches**
  - Became the **most common solution**
    - Current switches are **wire-speed** (non-blocking)
    - 802.1D – possibly with **multiple priority queues** (802.1p)
    - 802.1Q – **Virtual LANs**
  - Not perfect !
    - **Priority inversions** in queues (normally FIFO)
    - **Mutual interference** through shared memory and CPU
    - **Additional forwarding delay** (with jitter caused by address table look up, address learning, flooding)
    - Delays vary with switch technology and internal traffic handling algorithms

FlexRay

www.flexray.com

- Created in the early 2000s by an industrial consortium from the **automotive domain**.
- First public specifications available since 2004
- Is expected to replace CAN and contend with TTP/C !
- **Tx rate** up to **10Mbit/s**

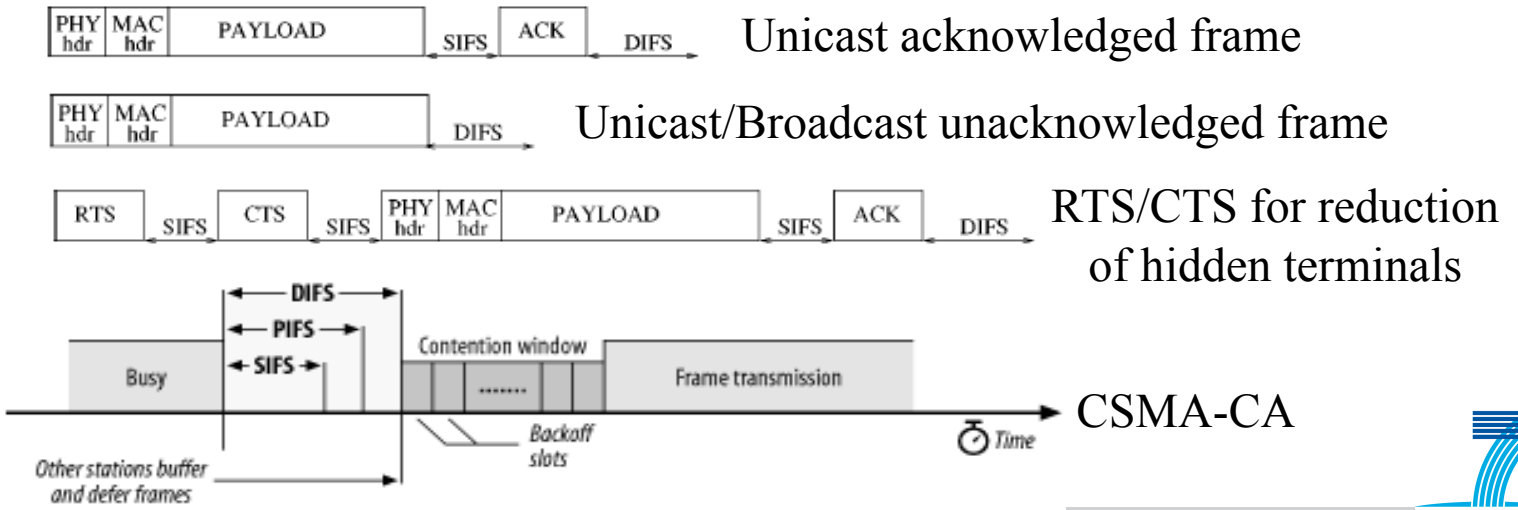




WiFi (802.11)

standards.ieee.org/getieee802/802.11.html

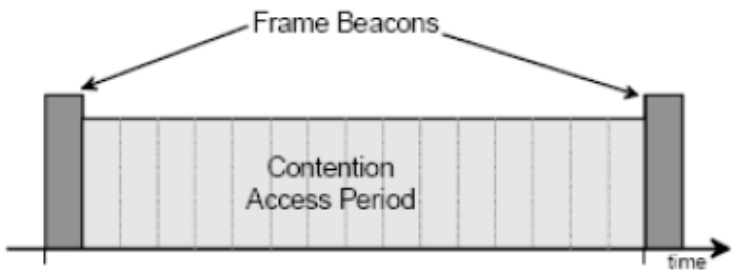
- Created in the 1990s for the SOHO domain and widely used for WLANs
- **3 modes/bands:** 802.11b/g (ISM-2.4GHz), 802.11a (5GHz)
- Scalable Tx rates between **1Mbit/s and 54Mbit/s**
- Many mechanisms to **reduce collisions** and **hidden-terminals**
- Growing interest for **QoS support** (initial PCF mode was abandoned, currently there is a new mode, 802.11e)



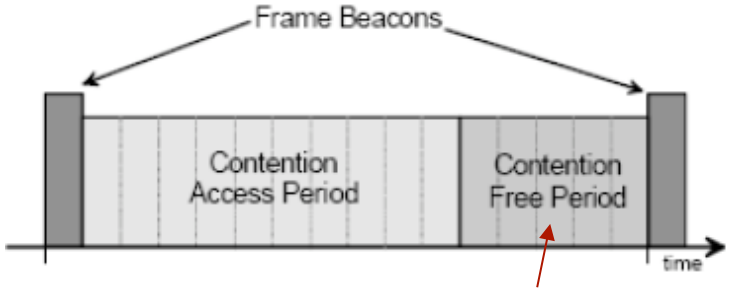
ZigBee

www.ZigBee.org

- Created in the early 2000s for **wireless monitoring and control**. Targets wireless sensor networks.
- Works on top of **802.15.4**, a PAN DLL. 3 bands: ISM-2.4GHz, 868/915 MHz. Targets **very low consumption**.
- Data rate of **250 Kbit/s** with range **up to 300m**
- **Beacon mode** and possible PAN coordinator to structure cells and QoS support. Structured and peer-to-peer modes. Routing support.
- Up to 65K nodes (full and reduced function devices)



Superframe without GTS
(Guaranteed Time Slot)

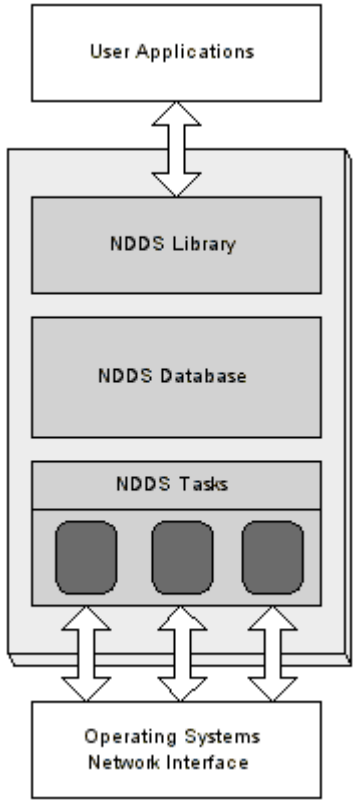


Superframe with GTS
(Guaranteed Time Slot)

DDS - Data Delivery Service

portals.omg.org/dds

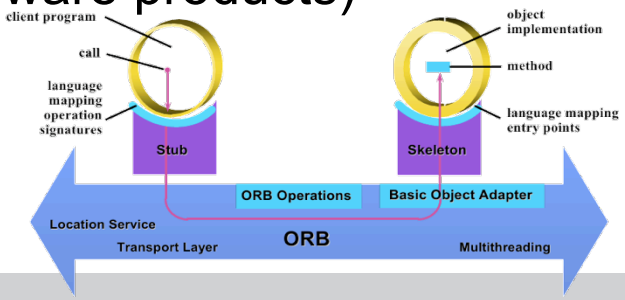
- **Real-Time Publisher-Subscriber** middleware for distributed real-time and embedded systems
- Standardized by **OMG** (Object Management Group)
- **DDS database shared among all nodes**, which have an holistic view on the communication requirements
- Publishers create "**topics**" (e.g. temperature) and publish "**samples**"
- Addressing, delivery, flow control, handled by DDS



CORBA - Common Object Request Broker Architecture

www.corba.org

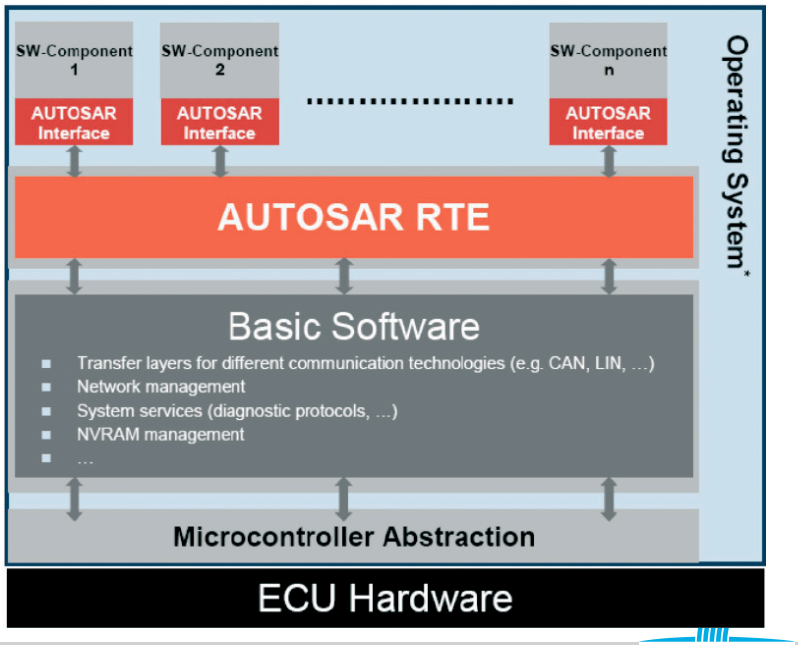
- **Open specification proposed by the OMG**
- **Objective**
  - Clients use remote objects as if they were local
- **Main features**
  - **Interoperability** between languages and platforms
    - Windows, Linux, Unix, MacOS, **QNX, VxWorks**, …
    - Ethernet, CAN, Internet, ...
    - Ada, C, C++, Java, Python, …
  - Multiple vendors (some are freeware products)
  - Many profiles
    - **Minimum CORBA, RT-CORBA**, FT-CORBA,



AUTOSAR

www.autosar.org

- **Proposed by a consortium of automotive industries**
- **Aims at separating functionality from execution HW**
  - Soft AUTOSAR components
- **Improve efficiency in using system resources**
- **Reduce number of active components and costs**
- **Manage complexity**
- **Give more design freedom to the OEM wrt subsystem providers**
- **Similar trends in avionics (IMA) and industrial automation (IEC 61499)**



Some open issues

Some open issues

- **In wired networks (DES or U/NES)**
  - Combining RT and nonRT nodes in a bandwidth efficient manner
  - Support virtualization in the network towards higher composability and robustness
    - Explore the use of stars to increase the robustness wrt errors and temporal misbehaviors and enforce partitions
  - How to guarantee non-functional properties with HW agnostic software components?
  - Adaptive QoS guaranteed channels (dynamic QoS management)
  - …

Some open issues

- **In wireless networks (U/NES)**
  - Improve synchronization for better RT behavior
    - For channel spatial reuse, wave-like data routing
  - MACs & routing to further improve energy savings
  - Virtualization in large scale networks
    - Resource reservation
  - The *hole problem* in routing
  - Resilience to interference
    - Improving frequency multiplexing
    - Exploiting new techniques, e.g., cognitive radio…
  - Improve bandwidth efficiency with new coding techniques
    - E.g., Network coding
  - …

Conclusion

- The **network is a fundamental component** within a distributed or networked system (supports **system integration**)
- **Real-time coordination** in a multi-node system requires **time-bounded communication**
  - appropriate protocols must be used
- We have seen a brief overview of the **techniques and technologies** used in the **networks and middlewares for embedded systems**
- Still many **open issues** remain in trying to **improve the timeliness**, **robustness** and **efficiency** of the communication