Redes de Comunicação em Ambientes Industriais Aula 3

Luís Almeida lda@det.ua.pt

Electronic Systems Lab-IEETA / DET
Universidade de Aveiro
Aveiro, Portugal

In the previous episode ...

- Distributed systems application domains and general concepts
- Requirements
  - Short data, short periods, low jitter, low latency
- Notion of message and transaction
- Temporal behavior merit figures
  - Network induced delay, delay jitter, buffers requirements, ...
- Data exchange semantics: event and state
- Time and event triggering

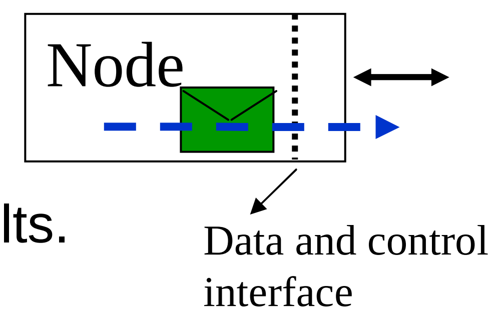
Outline

- Transmission control
  - who triggeres message transactions
- Information flow
- Operation flexibility
- Protocol stacks

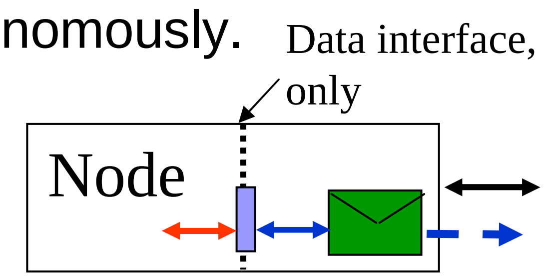
Transmission control

- 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.



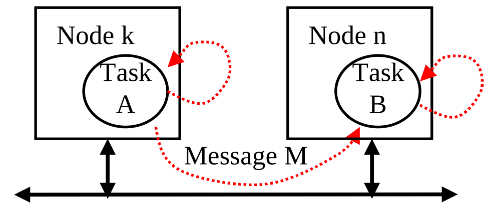
Transmission control

- All **4 combinations** of network type and transmission control are possible, depending on how much global a priori knowledge there is and whether it is located within or outside the network:
  - **ET network with external control**
    no global a priori knowledge
  - **TT network with autonomous control**
    network includes global a priori knowledge
  - **ET network with autonomous control**
    some global a priori knowledge within the network
    (e.g. predefined servers per node)
  - **TT network with external control**
    global a priori knowledge kept by the application but outside the network

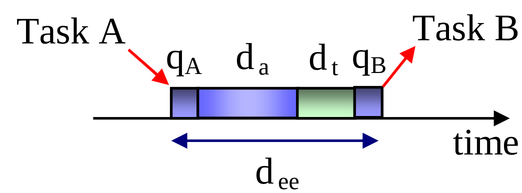
Typical approaches to combine ET and TT

Information flow

- ✓ Determining **end-to-end delay**

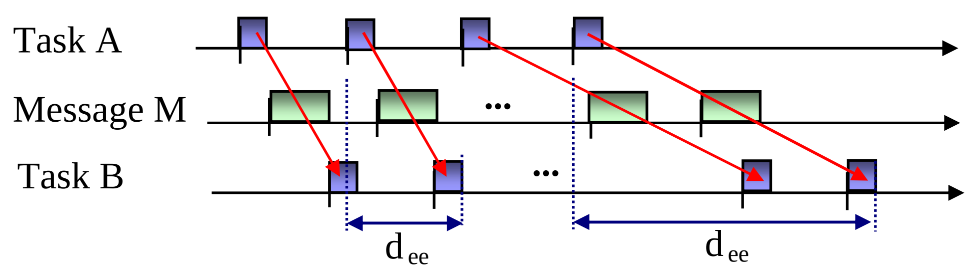


  - ✓ **ET-network with external control**
    Transactions are composed of several elementary actions carried out in sequence.



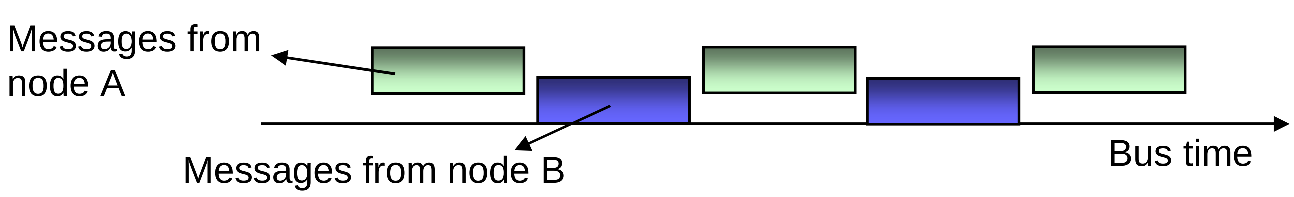
  - ✓ **TT-network with autonomous control**
    The elementary actions in each intervenient (transmitter, network, receiver) are decoupled, spinning at an appropriate rate.

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**



Operational Flexibility

- ✓ There is a growing interest in using Distributed Embedded Systems in **dynamic operational scenarios**:
  - ✓ Systems with variable number of users, either humans or not (traffic control, radar…)
  - ✓ Systems that operate in changing physical environments (robots, cars…)
  - ✓ Systems that can self-reconfigure dynamically to cope with hazardous events or evolving functionality (cars, planes, …)

Buzz words:

QoS adaptation, graceful degradation, survivability

Operational Flexibility

- ✓ Network requirement:
  Dynamic (flexible) management of bandwidth
  while guaranteeing real-time constraints
  - ✓ Act upon periodic communication, e.g. related to control information (potentially bandwidth consuming)
  - ✓ Adapt transmission rates according to effective needs
  - ✓ Explore subsystems that operate ocasionally
  - ✓ Explore variable sampling/tx rates according to the current system control stability state

Operational Flexibility

- ✓ **ET-networks are inherently flexible** wrt the communication requirements
- ✓ **TT-networks are not** because of a priori knowledge (normally static)

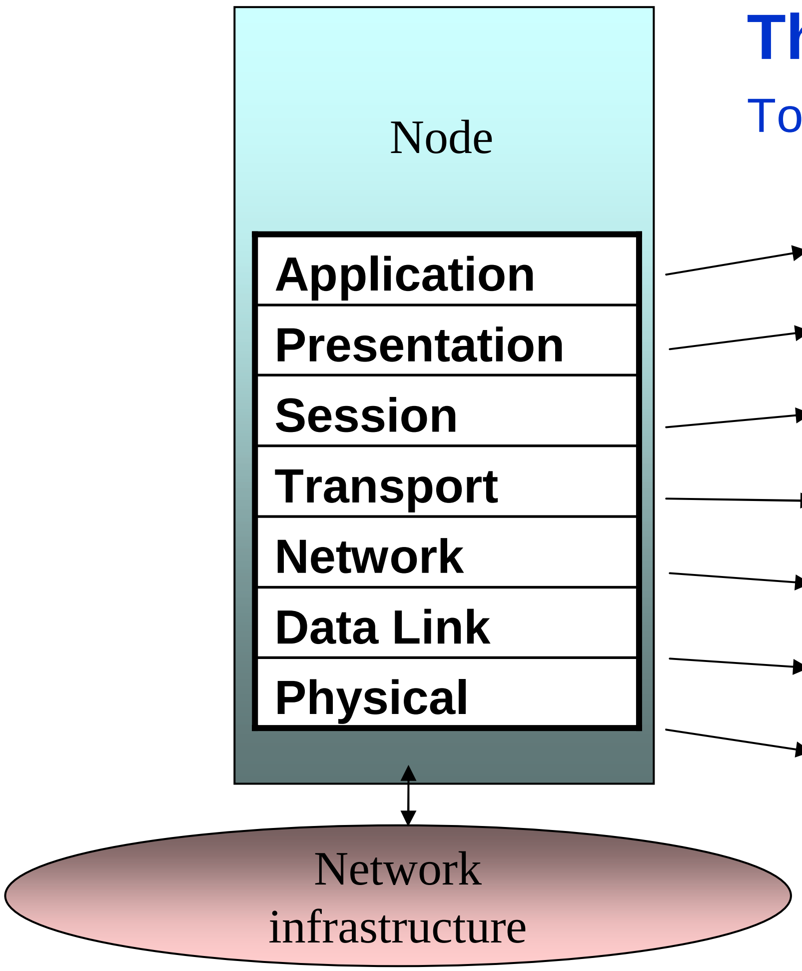
Question:

Are the TT-networks excluded when we need operational flexibility?

- ✓ **NO, flexibility of TT-networks** can be improved with:
  - ✓ **Multiple** operational **modes** that can be switched on-line
  - ✓ **On-line scheduling** of the **periodic traffic:**

→ **FTT paradigm** (**Flexible Time-Triggered** communication)

Protocol stack

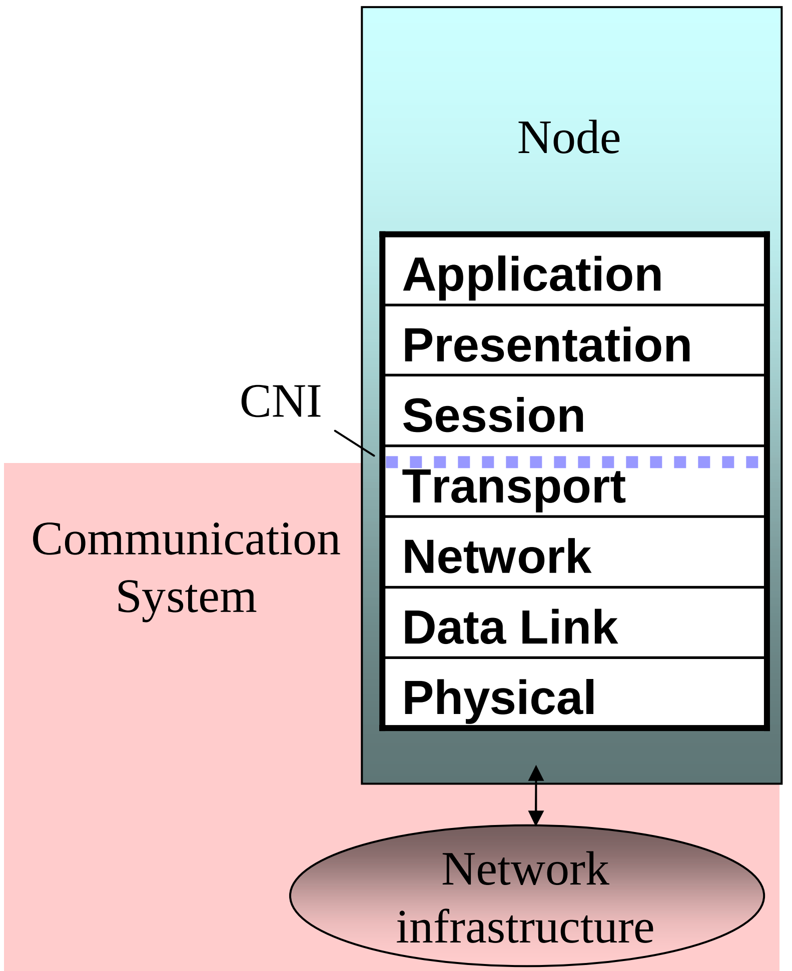


The OSI reference model

Towards interoperability and open systems

- Common services for class of apps.
- Data semantics
- Remote actions
- End-to-end comm. control
- Routing, logical addressing
- Physical addressing, MAC, LLC
- Topology, medium, bit-encoding

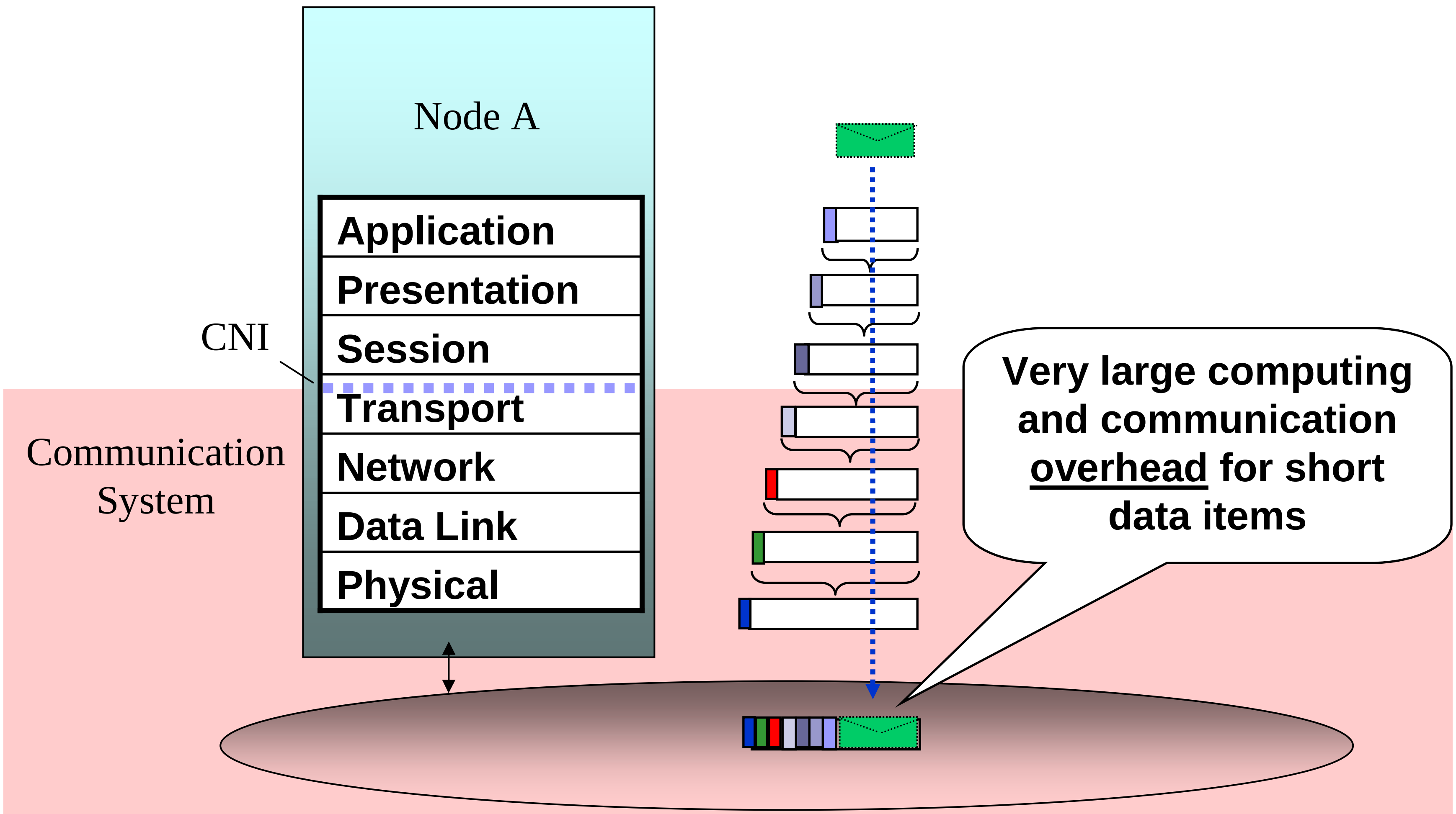
Protocol stack



Access to **information and communication services** for different types of applications

Services for the effective **transfer of information**

Protocol stack



Real-time protocol stack

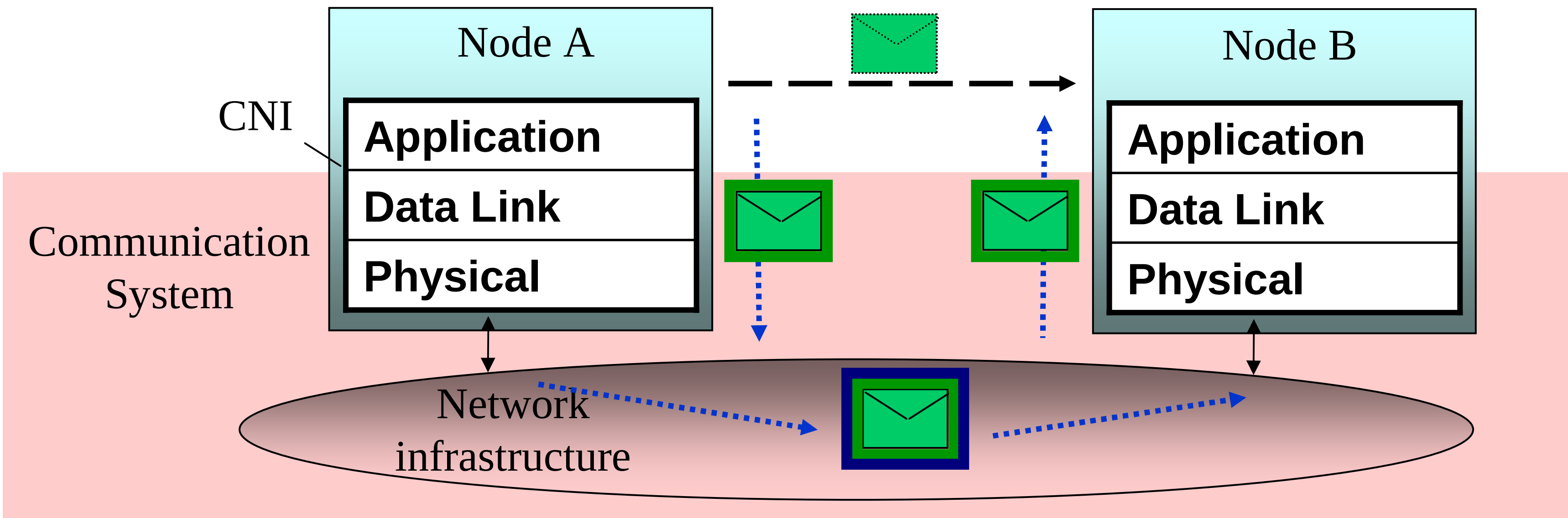
- The **end-to-end communication delay** must be **bounded**
  - **All services** at all layers must be **time-bounded**
  - Requires appropriate **time-bounded protocols**

- The **7 layers** impose a considerable computation and communication **overhead**…
  - The time to execute the protocol stack becomes dominant wrt the communication time

- Many 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
- Other layers maybe present but not fully stacked
- In process control and factory automation these networks are called **Fieldbuses**



Summary:

- Transmission **control** can be **external** or **autonomous**
- **TT networks** with **autonomous control** require **judicious use of relative phase** to avoid high delays and jitter → Time-triggered architecture
- **ET networks** are inherently **flexible** at run-time
- **TT networks** are typically **static** but can use multiple modes or on-line scheduling of the periodic traffic

Summary (cont.)

- The **OSI 7 layers** reference model imposes **too much overhead** for real-time networks (mainly in embedded control applications)
- **Real-time** properties must be enforced in **all layers**
- Real-time networks frequently use a **collapsed 3 layers** structure:
  - **physical**, **data link** and **application** layers