Redes de Comunicação em Ambientes Industriais
Aula 8

Luís Almeida lda@det.ua.pt

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

In the previous episode ...

- ✓ Cooperation models:
  - ✓ Client/Server, Producer/Consumer, Producer/Distributor/Consumer, Publisher/Subscriber
- ✓ Manufacturing Message Specification
  - ✓ Goals
  - ✓ Architecture
  - ✓ Objects and methods
- ✓ Clock synchronization
  - ✓ IEEE 1588
  - ✓ SynUTC

Traffic scheduling

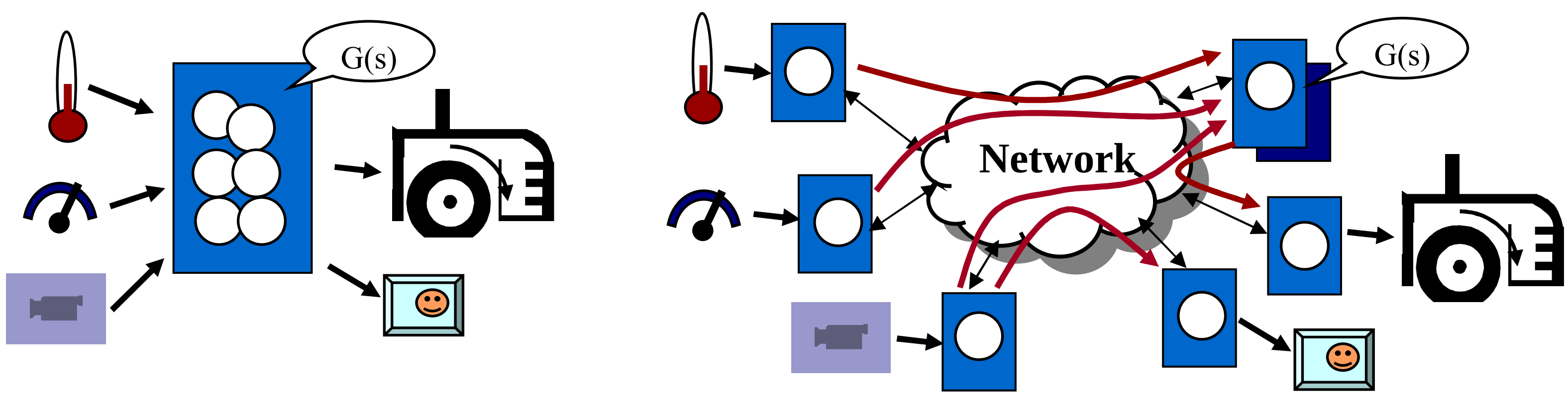
- Establishes the **relative order** of message transmissions
- Related issues:
  - **Constraints** imposed by the **MAC**
    Fixed-priorities, Master-Slave, Token-passing, TDMA, FIFO queues, Table-based
  - Support for **global synchronism**
    Allows use of offsets
  - **On-line** or **off-line** (table-based) scheduling
  - **Static** or **dynamic** scheduling

Traffic scheduling

- ✓ The **traffic scheduling** algorithm is essentially executed at the
  - ✓ **data link level** (**MAC** and by local **queuing** policies)
  - ✓ **network layer** (routing queues)
- ✓ It can be **distributed** (e.g. CAN), or **centralized** in a particular node (e.g. FTT-CAN, WorldFIP).

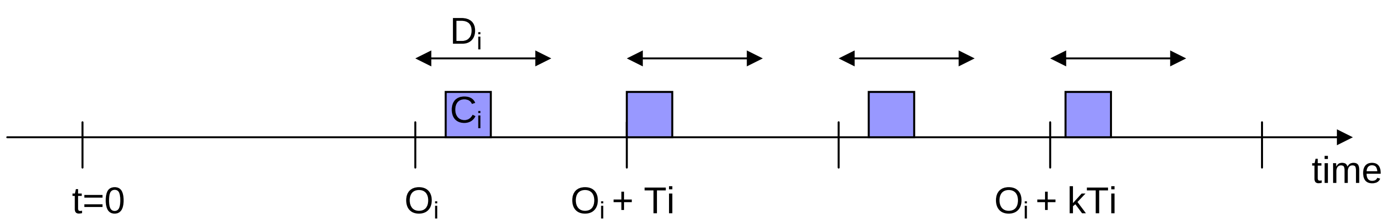
Traffic scheduling

- **Resemblances with task scheduling**
  - The problem of scheduling tasks in a processor, upon fully distribution (one processor per task) is transformed in a message scheduling problem
  - The **network** is now the ***bottleneck*** (i.e. the resource to be scheduled)



Traffic scheduling

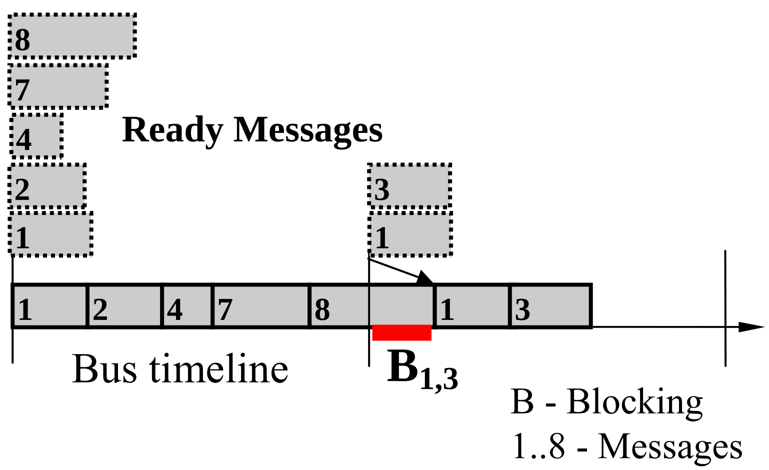
- **Resemblances with task scheduling**
  - **Task model must be adapted adequately according to network protocol**
    - Tasks **execution time** (**C_i**) translates to **message transmission time**, or to **transaction duration** when atomic
    - **Period** (**T_i**), **Deadline** (**D_i**) and **Priority** (**P_i**) are similar
    - **Offsets** (**O_i**) are supported on **globally synchronized systems, only**



Traffic scheduling

- ✓ **Resemblances with task scheduling**

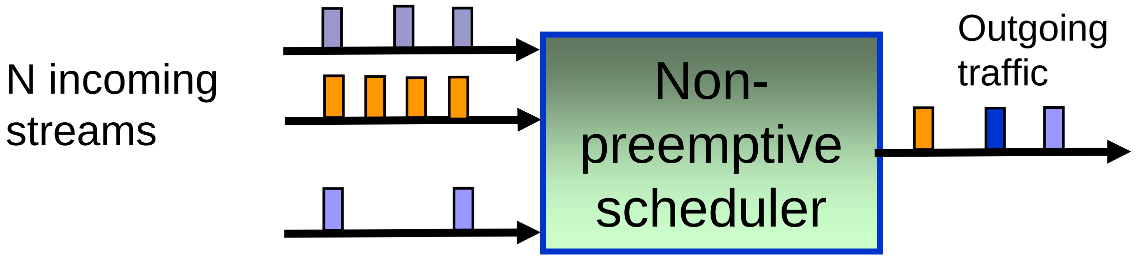
  - ✓ **Non-preemption** of message (packet) transmission. Once transmission starts cannot be interrupted. Can be accounted for as a **blocking term (B_i)**.



Traffic scheduling

- ✓ **Typical scheduling model**

$M \equiv \{m_i\ (C_i,T_i,J_i,D_i,P_i,O_i),\ i=1..N\}$



Problem: **Can all timing constraints be met?**
or **Is the message set schedulable?**

Schedulability Analysis

Knowing the **scheduling policy** and the **arrival pattern of the incoming** flows allows determining the **departing pattern** of the **outgoing flow**

Traffic scheduling

- ✓ **Scheduling Criteria**
  - ✓ **Fixed Priorities**
    - ✓ **Rate Monotonic (RM)**
    - ✓ Deadline Monotonic (DM)
    - ✓ *Importance*
  - ✓ **Dynamic Priorities**
    - ✓ **Earliest Deadline First (EDF)**
    - ✓ Least Laxity First (LLF)
    - ✓ First Come First Served (FCFS)

Traffic scheduling

- ✓ **Schedulability Analysis**
  - ✓ Most typical analysis focus on
    - ✓ **Bandwidth utilization** $U=\sum_{1}^{N}\frac{C_i}{T_i}$
    - ✓ **Network induced delay** also referred to as **worst-case response time** analysis.
  - ✓ In **static table-based** systems it is typical to use **branch and bound** techniques to optimize the schedule (e.g. wrt to jitter or precedences)

Traffic scheduling

✓ **Schedulability analysis based on utilization**

Rate Monotonic (RM):

$$\sum_{1}^{N} \frac{C_i}{T_i} + \max_{1..N}\left(\frac{B_i}{T_i}\right) < N(2^{1/N} - 1)$$

Earliest Deadline First (EDF):

$$\sum_{1}^{N} \frac{C_i}{T_i} + \max_{1..N}\left(\frac{B_i}{T_i}\right) < 1$$

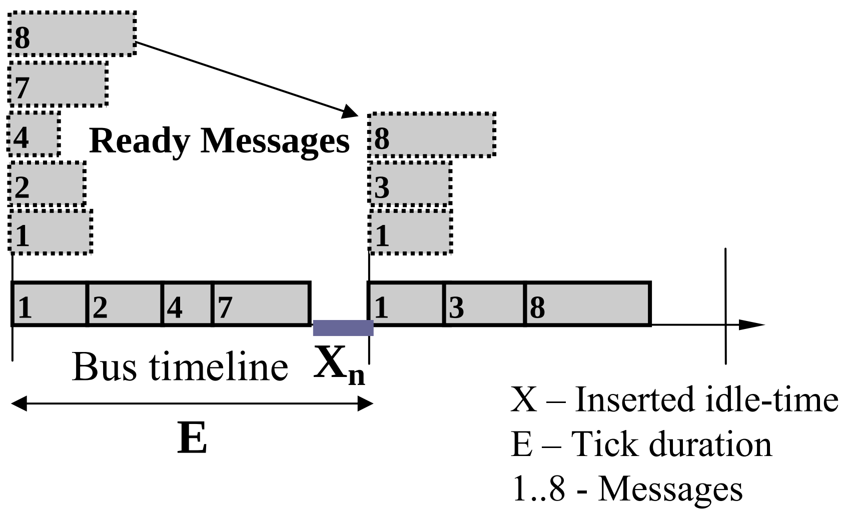
$$B_i = \max_{l=j..N} (C_l)$$

Modified Liu and Layland (1973) utilizations bounds accounting for blocking

Traffic scheduling

- ✓ **Schedulability analysis based on utilization**

  - ✓ Getting rid of the blocking with **inserted idle-time**



RM: $$\sum_{1}^{N} \frac{C_i'}{T_i} < N(2^{1/N} - 1)$$

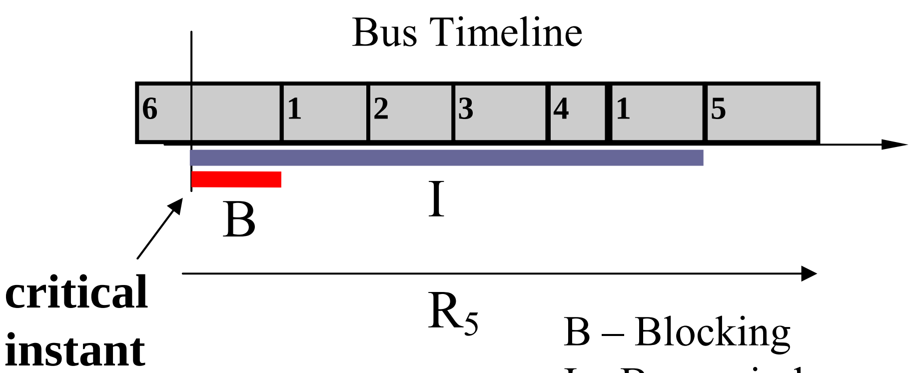
EDF: $$\sum_{1}^{N} \frac{C_i'}{T_i} < 1$$

$$C_i' = C_i * \frac{E}{E - X_{max}}$$

Non-Preemptive Blocking-Free Scheduling Model

Traffic scheduling

- ✓ **Schedulability analysis based on network-induced delay**
  - ✓ Maximum network-induced delay (Fixed Priorities)



B – Blocking
I – Busy window
R – Response time of 5
1..6 – messages

$$Rwc_i = I_i + C_i$$

$$I_i = B_i + \sum_{j \text{ in } hp(i)} \left\lceil \frac{I_i + \tau}{T_j} \right\rceil * C_j$$

Traffic scheduling

- ✓ **Schedulability analysis based on network-induced delay**
  - ✓ Maximum network-induced delay (Fixed Priorities)

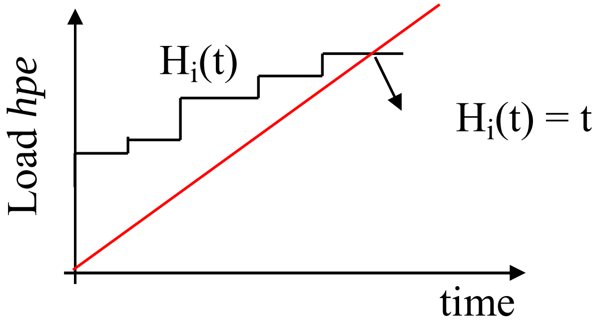
$$I_i(n+1) = B_i + \sum_{j \text{ in } hp(i)} \left\lceil \frac{I_i(n)+\tau}{T_j} \right\rceil * C_j$$

$$\text{until} \quad I_i(n+1) = I_i(n) \quad \text{or} \quad I_i(n+1) > D_i$$

$$\text{with} \quad I_i(0) = B_i + \sum_{j \text{ in } hp(i)} C_j$$

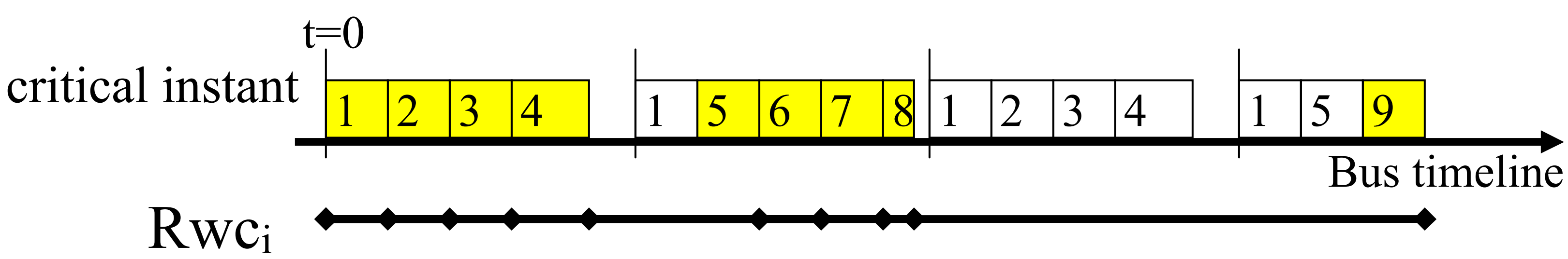
Load generated up to t

$$H_i(t) = B_i + \sum_{j \text{ in } hp(i)} \left\lceil \frac{t+\tau}{T_j} \right\rceil * C_j + C_i$$



Traffic scheduling

- **Schedulability analysis based on network-induced delay**
  - Maximum network-induced delay (Fixed Priorities)
  - With **inserted idle-time** we can use the **Timeline Analysis**
    - Consider the following set of 9 variables with periods given by $T_1=1$, $T_{2..5}=2$, $T_{6..9}>3$



Traffic scheduling

✓ Cyclic Table-Based Scheduling

- ✓ A table is built **off-line** with a **cyclic schedule**
- ✓ At run-time, the table is scanned to initiate transmissions according to schedule
- ✓ The table can be **centralized** (e.g. WorldFIP) or **distributed** (e.g. TTP/C)
- ✓ **Applicable to periodic traffic**

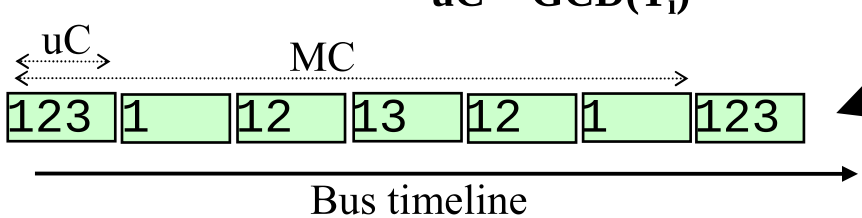
Periodic Messages:

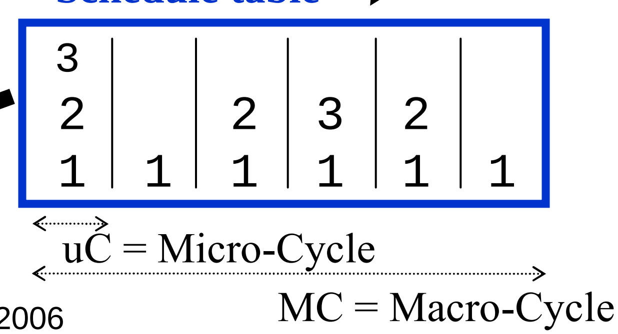
i	1	2	3
T_i	1	2	3

Messages table

$MC = LCM(T_i)$

$uC = GCD(T_i)$





Traffic scheduling

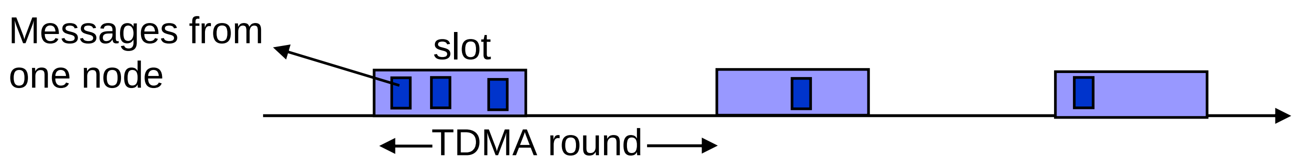
- ✓ **Cyclic Table-Based Scheduling**
  - ✓ Allows using **optimization techniques** (e.g. Branch and Bound, Simulated Annealing, Integer Linear Programming, Genetic Algorithms) to **improve schedule properties** (e.g. jitter, Rwc, precedences)

Traffic scheduling

- ✓ **Similarities with server scheduling**

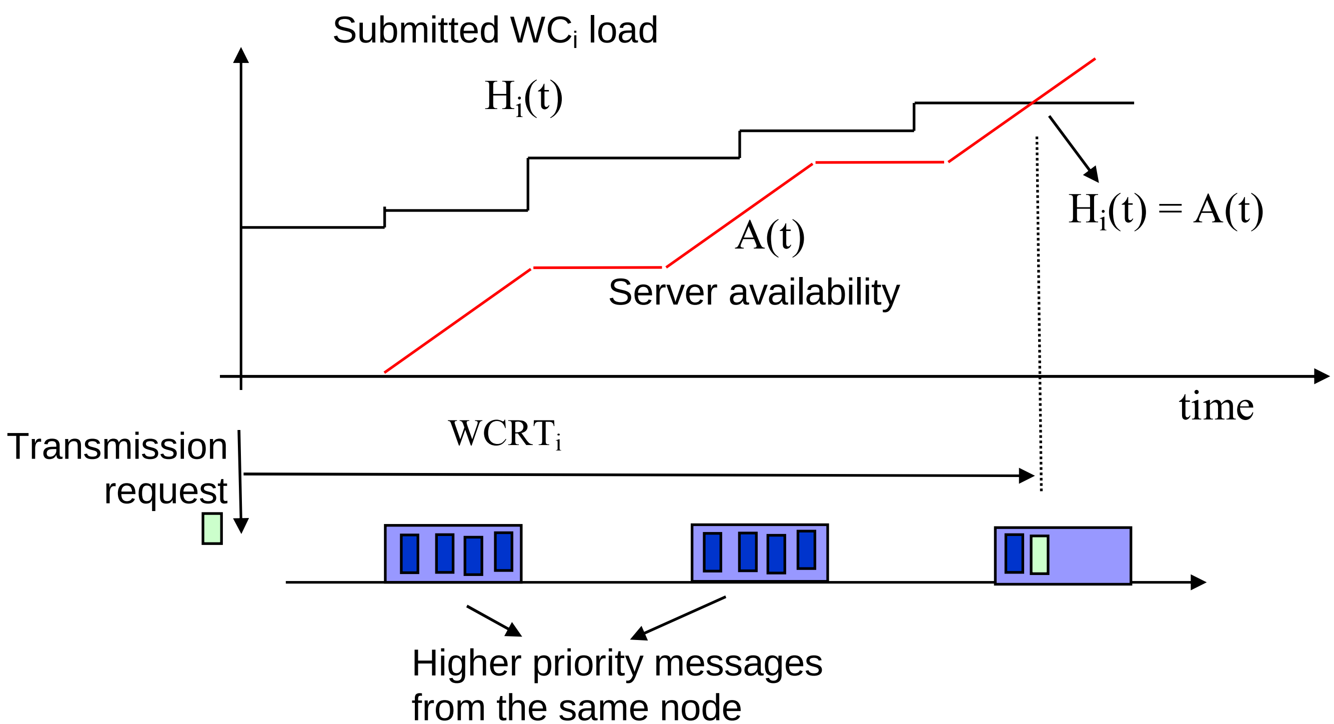
  - ✓ Typically, **controlled access networks** allocate a **fraction of bandwidth** (server) to each node. **Server-based analysis** for processor scheduling can also be used in this case, with adequate adaptations.

    e.g. a TDMA slot can be viewed as a server handling the traffic from the respective node.



Traffic scheduling

- ✓ **Similarities with server scheduling**



Traffic scheduling

- **Constraints imposed by the MAC**
  - **Minimum** transmission **period** (e.g. TDMA round cycle, or microcycle in Master-Slave.
  - **High jitter** in Token-Passing systems, due to the irregularity of token arrivals.
  - **Blocking** term in asynchronous systems (no offset, i.e. phase, control).
  - **Dead interval** in polling systems (e.g. Master-Slave, Token-Passing) to handle aperiodic communication requests.
  - **Inserted idle-time** in synchronous systems with variable size data.

Summary:

Traffic scheduling:

- ✓ Establishes the relative order of the message transmissions
- ✓ Carried out essentially at Data Link or Network layers
- ✓ Distributed/Centralized
- ✓ Resembles task scheduling (adaptation of the task model possible)

✓ Scheduling criteria:

- ✓ Fixed priorities (RM, DM, importance/value)
- ✓ Dynamic priorities (EDF, LLF, FCFS)

✓ Schedulability analysis:

- ✓ Utilization
- ✓ Response time
- ✓ Timeline
- ✓ Branch and bound (for static table/based)

Summary:

- Similarities with server scheduling
  - fraction of the network bandwidth allocated to each node
- MAC imposes constraints
  - minimum transmission period
  - jitter
  - blocking
  - dead-interval
  - inserted idle-time