Redes de Comunicação em Ambientes Industriais
Aula 7

Luís Almeida lda@det.ua.pt
Paulo Pedreiras pedreiras@det.ua.pt

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

In the previous episode ...

The Data-Link Layer

- **Addressing:**
  - **Direct and indirect (source, time-based)**
- **Logical link control – LLC**
  - **Services: send with(out) ack., request data, connection-oriented**
  - **Transmission error control : Forward Error Correction (FEC), Automatic Repeat reQuest (ARQ), Positive Acknowledge and Retry (PAR)**
- **Medium access control – MAC (for shared medium)**
  - **master/slave, token passing, TDMA, CSMA/CD, CSMA/BA(CA), micro-segmentation**

Application layer

- ✓ Issues related with the application layer:
  - ✓ Cooperation models
    - ✓ Client-Server
    - ✓ Producer-Consumer
    - ✓ Producer-Distributor-Consumer
    - ✓ Publisher-Subscriber
  - ✓ MMS
  - ✓ Clock synchronization
    - ✓ IEEE 1588
    - ✓ SynUTC

Application layer

- ✓ Cooperation model - **Client-Server**
  - ✓ Transactions are **triggered** by the **receiver** of the requested information (**client**).
  - ✓ Nodes that **generate** information are **servers** and only react to client requests.
  - ✓ The model is based on **unicast** transmission (one sender and one receiver)

Application layer

- ✓ Cooperation model – **Producer-Consumer**
  - ✓ Transactions are **triggered** by the nodes that **generate** information (**producers**).
  - ✓ The nodes that **need** the information, identify it when transmitted and retrieve it from the network (**consumers**)
  - ✓ The model is based on **broadcast** transmission (each message is received by all nodes)

Application layer

- Cooperation model – **Producer-Distributor-Consumer**
  - Basically similar to Producer-Consumer
  - Transactions are **triggered** by a particular node, the **distributor**, upon request from the producers or according to a pre-established schedule.
  - It is an implementation of PC over master-slave

Application layer

- ✓ Cooperation model – **Publisher-Subscriber**
  - ✓ Elaborate version of Producer-Consumer using the 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**).

MMS

- Manufacturing Message Specification
  - OSI application layer messaging protocol
  - Exchange of real-time data and supervisory control information between networked devices and applications
  - Defines common functions for ditributed automation systems
  - Originaly published in 1990 by ISO TC 184, (application layer of GM MAP)
  - Standardized as ISO 9506-1/2
  - Nowadays implemented in a wide range of networks (Ethernet, fieldbuses, RS485, ...)

MMS

- ✓ Manufacturing Message Specification
  - ✓ Object-oriented modelling
  - ✓ Based on Client-server model

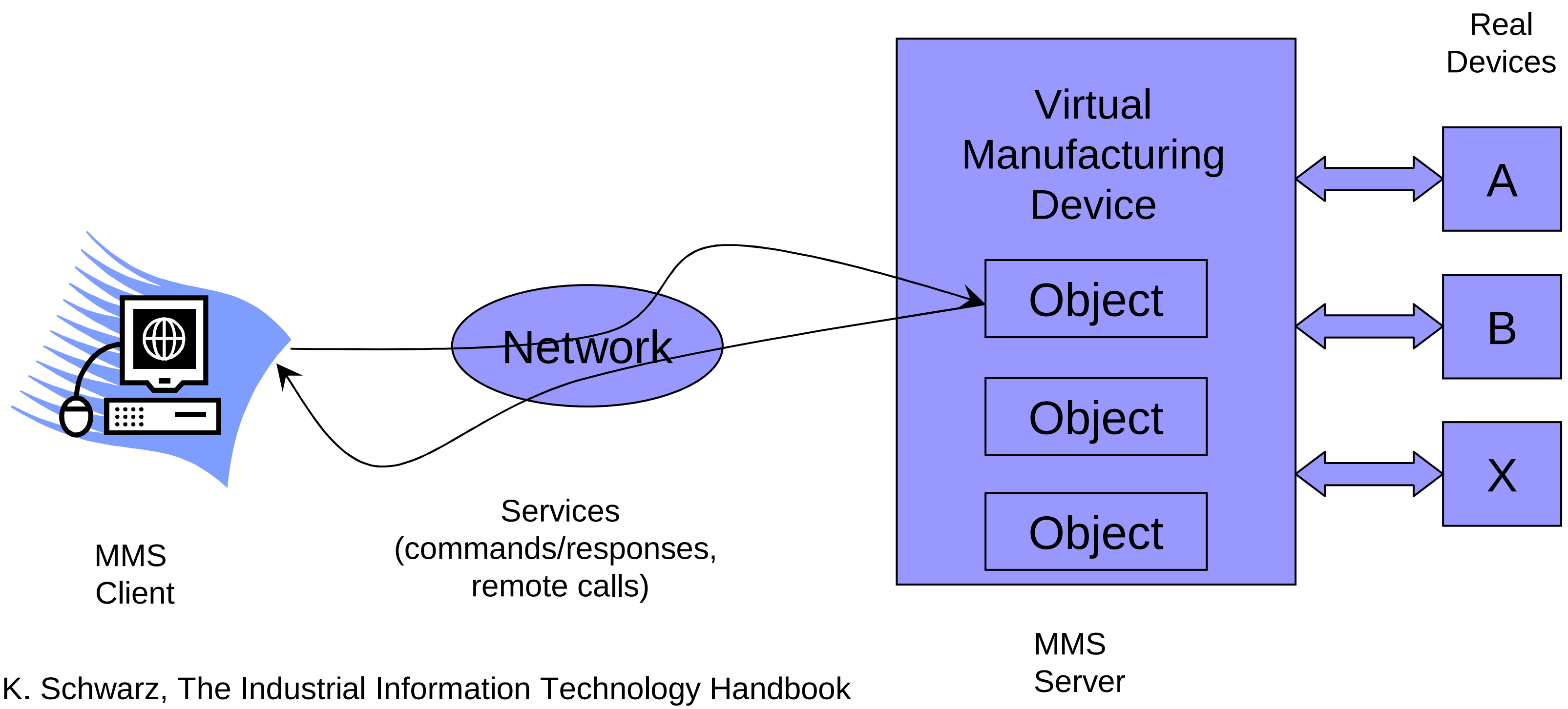

Fig. K. Schwarz, The Industrial Information Technology Handbook

MMS

- Object classes supported
  - Named Variable
  - Domain
  - Named Variable List
  - Journal
  - Semaphores
  - ...
- Methods (remote calls, commands)
  - Read/write
  - Information report
  - Download
  - Read journal
  - ...

MMS

- ✓ Real data representation and implementation details are completely hidden
- ✓ MMS does not define implementation details; only:
  - ✓ i) how the objects behave and represent themselves to the outside
  - ✓ ii) how clients van access objects
- ✓ Goals:
  - ✓ Interoperability
  - ✓ Independency of device manufacturers, network, hardware architecture, ...

MMS

✓ MMS objects (1)

- ✓ **VMD.** The device itself (Function, vendor, model, ...)
- ✓ **Domain.** Represents a resource (e.g. a program, a memory region) within the VMD
- ✓ **Program Invocation.** An executable program consisting of one or more domains
- ✓ **Variable.** An element of typed data (e.g. integer, floating point,...)
- ✓ **Type.** Format of a variable's data.
- ✓ **Named Variable List.** A list of variables that is named as a list
- ✓ **Semaphore.** To control access to shared resources

MMS

✓ MMS objects (2)

- ✓ **Operator Station.** An operator display and keyboard
- ✓ **Event Condition.** An object that represents the state of an event
- ✓ **Event Action.** Represents the action taken when an event condition changes state
- ✓ **Event Enrollment.** Which network application to notify when an event condition changes state.
- ✓ **Journal.** A time based record of events and variables.
- ✓ **File.** A file in a filestore or fileserver.
- ✓ **Transaction.** Represents an individual MMS service

MMS

- ✓ **MMS objects and services**

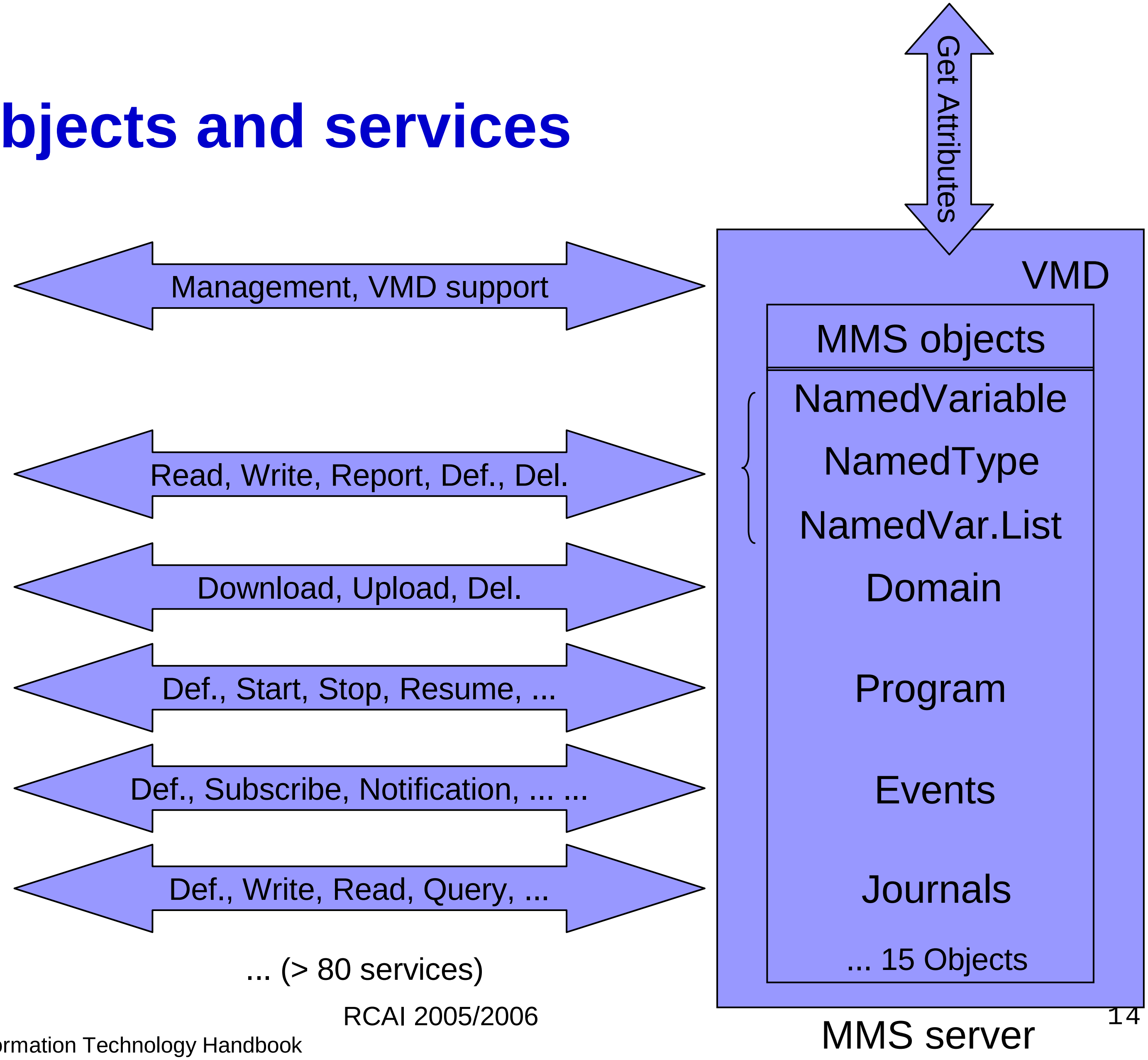

Fig. K. Schwarz, The Industrial Information Technology Handbook

Clock synchronization

- ✓ Distributed systems need to have a common notion of time 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
  - ✓ etc.

Clock synchronization

Synchronization requirements

- ✓ Generic data networks:
  - Applications: distributed file systems, financial transactions, office applications
  - Accuracy: from milliseconds to seconds
  - Protocols: Network Time Protocol (NTP) (covers the LAN and WAN area)
- ✓ Distributed real-time systems:
  - Applications: supervision, measurement and control systems
  - Accuracy: from sub-microseconds to milliseconds
  - Protocols: IEEE1588, SynUTC, ...

Clock synchronization

- ✓ Example: Substation Automation

  Class/Sync. accuracy: T1/1 ms ; T2/0.1 ms ; T3/±25 µs ; T4/±4 µs ; T5±1 µs

- ✓ Additional requirements for industrial networks:
  - ✓ Must be available on different networking technologies (not only Ethernet)
  - ✓ A minimum of administration is highly desirable,
  - ✓ The technology must be capable of implementation on low cost and low-end devices,
  - ✓ The required network and computing resources should be minimal.

Clock synchronization

IEEE1588 overview (1)

- Hierarchic, master/slave
  - *grandmaster clock:* best clock in the system
  - *subnet master:* best clock in a subnet

  (single subnet: grandmaster and master are the same)
- In each subnet nodes synchronize with the subnet master
- Subnet master synchronize with the grandmaster
- Master election is automatic (Best Master Clock algorithm). Mechanisms for indication a preferred set of masters
- External synchronization possible (e.g. GPS)

Clock synchronization

IEEE1588 overview (2)

- Operation:
  - Master clocks periodically send timing messages to slaves
  - Slaves send timing messages to masters for automatic calibration of communication latency.
- Attributes:
  - On average one packet/second (low communication overhead)
  - Minimal computing and memory resources are required (Implementation in simple uC devices possible)
- Accuracy:
  - Sub-microsecond using hardware-assist techniques (referenced in the standard)
  - Tens of microseconds or more with software-only implementations (interrupt-driven or kernel-level code)

Clock synchronization

IEEE1588 overview (3)

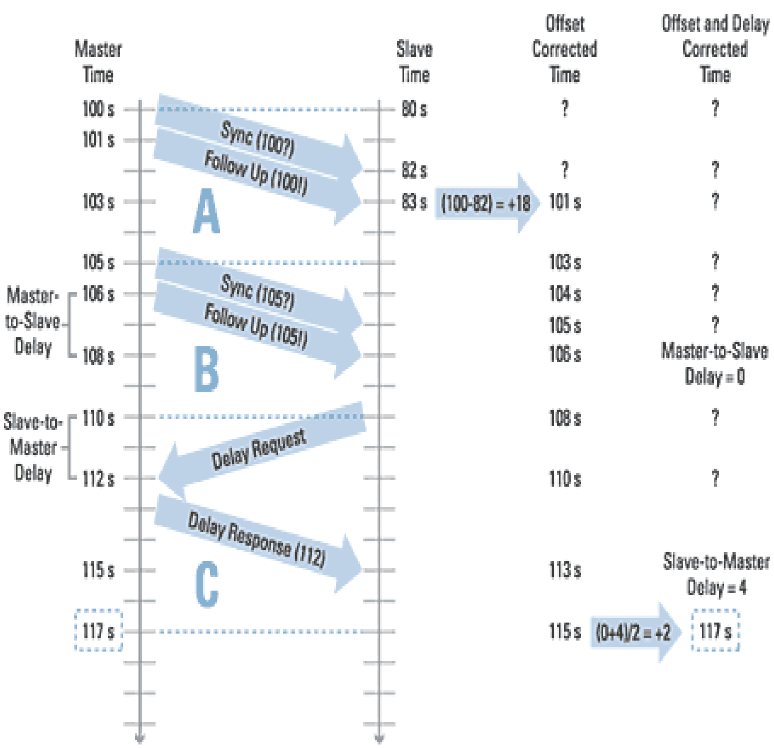

Figure: “Special Focus: Understanding the IEEE 1588 Precision Time Protocol, National Instruments”

1 - Master sends a sync message

2 - Slave timestamp arrival of the sync message uses its local clock to and compare it with the actual sync transmission timestamp carried out in the master clock’s follow-up message.

3-The difference between the two timestamps represents the offset of the slave plus the message transmission delay, which is used to adjusts the slave local clock at point A.

4 – The process is repeated to account for message jitter

5 – Slave send a Delay Request to the Master. The master gets the reception instant and send it to the slave in a Delay Response message. On reception the slave computes the slave-to-master delay and updates the local clock, which is now synchronized.

Clock synchronization

SynUTC (Synchronized Universal Time Coordinate)

- ✓ Developed at the Vienna University of Technology
- ✓ Fault-tolerant high accuracy time synchronization (sub-microsecond)
- ✓ Aims specifically at Ethernet LANs
- ✓ Based on specific hardware, both at the end-nodes and switches

  packet time-stamping carried out in hardware (MII – media independent interface)

Clock synchronization

SynUTC technology:

- ✓ Adder-based clock: fine-grain adjustable (nsec/sec steps)
- ✓ On-the-fly time-stamping:
  - Transmission: “Transmit TS” field automatically updated when the message begins to be transmitted
  - Reception: time-stamp placed on the “Receive TS” field when an Ethernet SFD delimiter is observed
- ✓ Interval-based paradigm:
  - ✓ Local clocks continuously maintain an accuracy interval:

$$Cp(t)-\alpha^{-}(t) \leq t \leq Cp(t)+ \alpha^{+}(t)$$

Clock synchronization

- ✓ Fault-tolerant average (FTA) algorithm
- ✓ Distributted architecture
- ✓ Periodically nodes exchange state and rate data
- ✓ Each node assembles a set of remote accuracy intervals and remore rate intervals
- ✓ The compute new local accuracy and rate intervals and
- ✓ Adjust local oscillator-clock parameters

Summary:

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