Intelligent distributed systems

Daniele Fontanelli

Department of Industrial Engineering
University of Trento
E-mail address: *daniele.fontanelli@unitn.it*

2022/2023

Outline

1. The big picture
    - Terminology
    - In this course...
    - Useful definitions for distributed systems
2. Take home message

Outline

1. The big picture
   - Terminology
   - In this course...
   - Useful definitions for distributed systems
2. Take home message

The ingredients

The basic components of a distributed system are:

- A set of (possibly heterogeneous) *plants* to be controlled;
- A set of input devices (*sensors*) and output devices (*actuators*) deployed in the environment;
- A set of *precessing units* implementing the system control;
- A set of (possibly heterogeneous) *communication* links.

Outline

1. The big picture
   - Terminology
   - In this course...
   - Useful definitions for distributed systems
2. Take home message

Terminology

Distributed control systems

Distributed Control Systems (DCSs) are control systems in which the controller elements are not geographically central but are distributed throughout the system.

They comprise a set of components, i.e., *sub–systems*, controlled by one or more controllers.

The entire system of controllers is connected by networks for communication and monitoring, i.e., sensing.

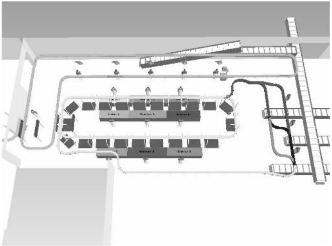
The elements of a DCS may connect directly to physical equipments, i.e., actuators.

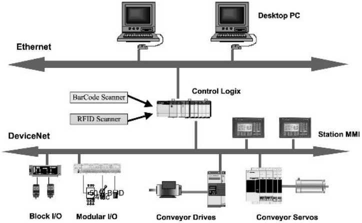
Due to technology push, DCS are becoming very close to the *Supervisory Control And Data Acquisition* (SCADA) systems, which will be also covered in this course.

We will refer, in general, to *Distributed Systems* as DCSs, SCADA, teams of robots, teams of sensors or combinations thereof.

Examples

Industrial plants

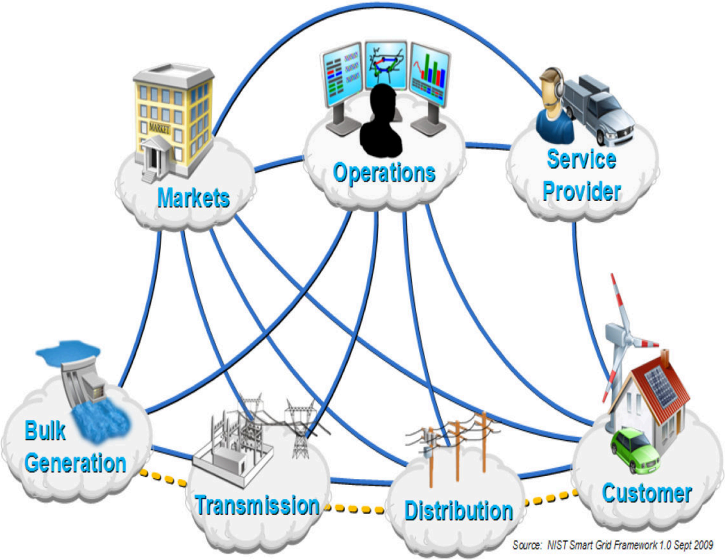




A. Mousavi, M. Sarhadi, A. Lenk, S. Fawcett, "Tracking and traceability in the meat processing industry: a solution", British Food Journal

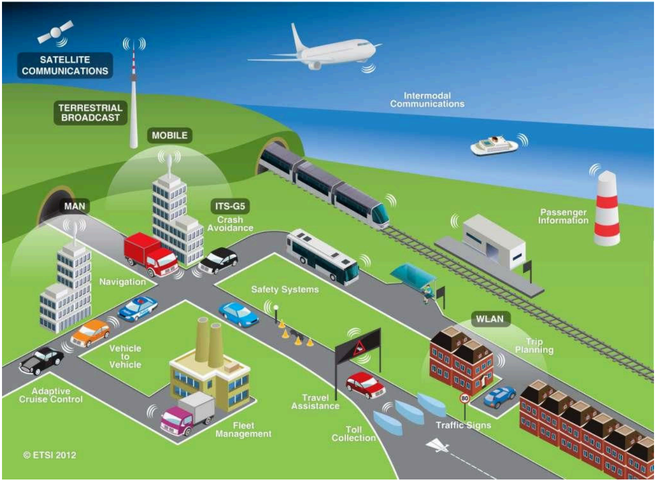
Examples

Smart grid



Examples

Road transportation systems



Outline

1. The big picture
   - Terminology
   - In this course...
   - Useful definitions for distributed systems
2. Take home message

In this course...

Distributed control systems

In this course, large part of the course will be focused on the *industrial and robotic applications*.

As a consequence, the DCS can be viewed as:

Definition

The DCS is a *complex physical system* that is interacting with a number of sub-systems for monitoring, control and management.

Outline

1. The big picture
   - Terminology
   - In this course...
   - Useful definitions for distributed systems
2. Take home message

Useful definitions

Real-time what??

Definition

A *real-time* computer system is a computer system in which correctness of the system behaviour depends *not only* on the *logical results of the computations*, but also on the *physical instants at which these results are produced*.

H. Kopetz, "Real-Time Systems, Design Principles for Distributed Embedded Applications." Kluwer Academic Publishers, Boston, Dordrecht, London, 1997.

Hence the computation should be *predictable in time*.

Real-time systems

Enabling tools of an RT system:

- *Communication and synchronisation* between the different application components;
- *Predictable response* to events (e.g., clock triggers, availability of sensor data, etc) and reliable input/output (I/O);
- *Prompt resource management* (e.g., computing power, sensor/actuator access, memory allocation, file synchronisation etc).

Real-time

Definitions

Real-time (RT) applications provide an action or an answer to an external event in a timely and predictable manner.

RT *does not* mean fast, since time scale may differ for different applications.

For example, controlling a vehicle engine requires a time scale of some microseconds, while temperature control in a HVAC acts on the tens of seconds time scale.

In both cases, the success of both applications depends on well-defined *time requirements* and, hence, may rely on largely different resource requests.

Real-time

Definitions

The concept of *predictability* is wide, however in RT applications it generally means that a certain operation or application can always be completed within a *predetermined amount of time*.
An unpredictable RT control application generates loss of sensing and actuation, hence may lead to faults and/or system instability.
In industrial applications two standards have been historically considered for real–time description: *IEC 61131-3* and *IEC 61499*.

Useful definitions

Real-time

Execution time

How can be ensured predictability of executions?

To have control over the predictability of an application, the engineer should take care of the *time bounds*, i.e. the *deadlines*.

For example, considering control applications running on *Embedded COmputing Unit* (ECU), the *time bounds* can be defined as the average *execution time* of an application, which leads to probabilistic guarantees.

Otherwise, the *Worst Case Execution Time* (WCET) can also be considered.

Useful definitions

Useful definitions

Real-time

Event vs Time triggered

Two different paradigms are prevalent for real-time architectures:

- In *event-triggered* systems the real–time actions are *triggered* by an environmental *event* (usually, a measure performed by a sensor).
- In *time-triggered* systems the real–time actions are triggered by the progression of a *global* time. Notice that if the system is distributed, the time must be *common*, i.e. the components should be *synchronised*.

Real-time

Event vs Time triggered

Definition

An *event* is a change of state, occurring at an instant.

Definition

An *event trigger* is a control signal that is derived from an event, i.e. a state change in a real-time entity.

Definition

A *time trigger* is a control signal that is generated at a particular point in time of a synchronised global time base.

Real-time

Event vs Time triggered

The major difference between event-triggered and time-triggered approaches is in *how* the actions are fired.

Time-triggered systems are in essence *autonomous* and interact with the environment according to an internal predefined *schedule*.

Smart grid example: estimates of the grid state.

On the other hand, event-triggered systems are under the control of the environment and must respond to its stimuli whenever they occur.

Smart grid example: reaction to blackouts.

Real-time

Event vs Time triggered

Due to the inherent higher predictability, the time–triggered is far more adopted in industry and robotics applications.

On the other hand, the event-triggered control paradigm may be preferred due to *higher flexibility and better resource exploitation*.

In event-triggered architectures, the *resources* (i.e., computing power, network bandwidth) are allocated only when needed. Hence, event-triggered approaches support *dynamic resource allocation strategies and resource sharing*.

Useful definitions

Real-time

I/O throughput

Is there any index quantifying the speed of computation or communication for a digital system?

Definition (Merriam-Webster dictionary)

The *throughput* is the amount of something (such as material, data, etc.) that passes through something (such as a machine or system).

For *digital systems*, this quantity is generally measured as *operations per second* for computing systems or as *transmitted bits per second* for communication systems and it is a function of the *bit rate*.

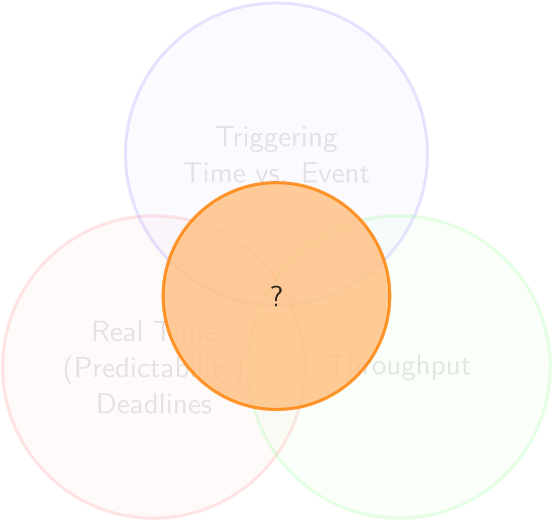
Real-time

I/O throughput for DCS

For DCS we have to take into account the presence of (possibly multiple) communication channels, which have their own characteristics and throughput.

However, it is evident from this discussion that a communication system that is able to ensure *predictability* of communications, hence to satisfy RT requirements, and a high *throughput*, hence to transfer the desired amount of data timely, is what we are looking for a DCS system.

Useful definitions



Real-time

Delays

Let us recall a standard output $y(t)$ feedback system, in which the control inputs are computed as:

$$u(t) = f(y(t)).$$

In a DCS there can be a communication channel between the output given by the available sensors and the system computing the control input $u(t)$. Moreover, the control input is usually computed by an embedded computing system, which will take some time to accomplish the desired computation.

Real-time

Delays

Therefore, we may model this unavoidable effect by:

$$u(t + \delta t) = f(y(t)),$$

The more δt is prominent with respect to the system time-scale, the higher will be the discrepancy between the hypothesised behaviour and the real behaviour.
This effect is broadly known in Automatic Control Theory with the name *input/output delay*.
For communication systems, it is called *end to end delay*.
But it can be even worse...

Real-time

Delays

If the delay δt is known upfront, it can be reasonably considered and compensated in the controller design, for example by adding a *unitary delay* on purpose to the plant to be controlled or by designing a properly robust control.

Yes, but what happens when the delay *changes in time*? And, what happens when it is *unpredictable*?

This is a known problem in the literature, since it is present in computing systems as well as in communication systems...

Real-time

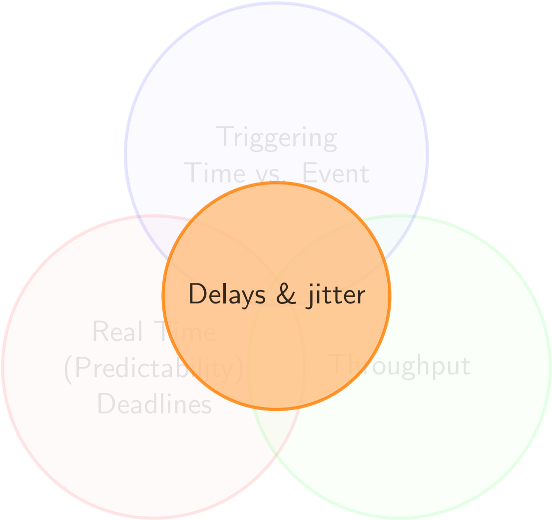
Jitter

Definition

The *Jitter* is defined as the deviation of a periodic signal from pure periodicity.

The meaning of jitter is *context dependent*. For digital communication networks, it is the *variation in latency* as measured in the variability over time of the packet delay across a network.
For computing systems, it is the variability of the time incurring *between the start of the computation and its end*.
Under the presence of the jitter, a stable system may easily become *unstable*, hence, we have to add this problem in the picture derived so far for DCS...

Useful definitions



Outline

1 The big picture
- Terminology
- In this course...
- Useful definitions for distributed systems

2 Take home message

Take home message

The basic ingredients of a DCS are:

- Several (digital) systems *interacting together* to obtain a certain goal;
- Each component can be seen as a *control* and/or a *sensing* system acting *locally* but having a *global* result;
- A DCS system comprises *plants*, *actuators*, *processing units* and a *communication system*.
- Both processing units and communication systems need *Real-Time* capabilities, a *triggering* scheme, an adequate *throughput* and has to deal with *jitter*.