Intelligent distributed systems

Daniele Fontanelli

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

2022/2023

Outline

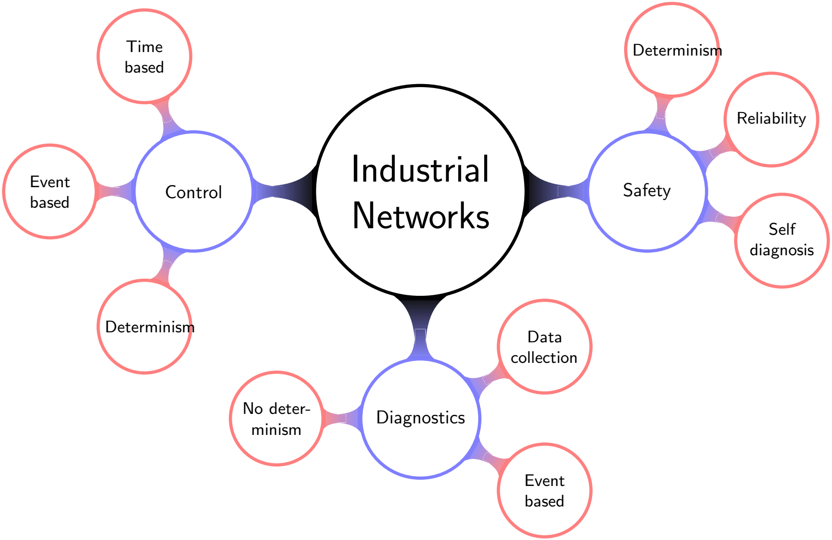
1. Trends in manufacturing networks
   - Parametrisation of Industrial Networks
   - Transmission times
   - Delays and jitters
2. Take home message

Outline

1. Trends in manufacturing networks
   - Parametrisation of Industrial Networks
   - Transmission times
   - Delays and jitters
2. Take home message

Industrial networks

Subdomain of applications



Outline

1. Trends in manufacturing networks
   - Parametrisation of Industrial Networks
   - Transmission times
   - Delays and jitters
2. Take home message

Network parametrisation

Although any network protocol can be used to send data, each network protocol has its pros and cons.
The main questions to answer when selecting a network are:

- Will the network carry *many small packets* of data frequently or *large packets* of data infrequently?
- Must the data arrive before a given *deadline*?
- *How many nodes* will be competing for the bandwidth, and how will the contention be handled?

Network parametrisation

Some possible choices

Two possible parameters that can answer to the previous questions are the *average data rate* and the *determinism*.

The *average data rate* is function of the network *access time* and *bit transfer rate*, i.e., the *throughput*.

The *determinism* is a measure of the ability to communicate data consistently from end-to-end within a guaranteed time.

As we saw, parameters that affect determinism are the *Physical layer* adopted, e.g., wired or wireless, the *MAC sublayer* and, partially, all the other layers implemented in the ISO OSI model.

Network parametrisation

Some possible choices

Another important feature for an industrial distributed application is the *device data rate*, in terms of *reaction* to events and *computation* times. Indeed, is becoming more and more evident in practical applications that the device data rate plays a *major* role in the plant effectiveness.
The lowest between all *the different* network segments and the device performance is the *communication bottleneck*.

Network parametrisation

Quality of Service

Is there any unified vision about this topic?
The *Quality of Service* (QoS) of a network is a *multidimensional parameterised measure* of how well the network exports its functionalities.
Associated to the QoS there are the capability of *evaluating* and *controlling* its parameters.
The QoS is critical since there is a *shared resource* that has to be mediated among the different network nodes in order to give guarantees about the time to deliver correctly the information.

Network parametrisation

Quality of Service

QoS

How much data?	*Data rate* and *throughput*
Time characteristics?	*Latency* and *jitter*
Network characteristics?	*reliability* and *security*

For *control networks*, it is of prominent importance to assess *determinism* as a QoS parameter.

Outline

1. Trends in manufacturing networks
   - Parametrisation of Industrial Networks
   - Transmission times
   - Delays and jitters
2. Take home message

Quality of Service

Data Rate

Recall that the *data rate* of an industrial network is given in terms of the number of bits that can be transmitted per second.

Industrial Networks data rate

Type	Max. data rate
CAN-based networks	1 Mb/s
Ethernet-based networks	1 Gb/s, up to 10 Gb/s (*optical fibre*)
DeviceNet	500 kb/s

A lot of networks currently used in the manufacturing industry are based on 10 and 100 Mb/s *Ethernet*.

Quality of Service

Data Rate

The *time to transmit a bit* is the “inverse of the data rate”.

Time to transmit one bit

Type	Data rate	Bit time t_{bit}
CAN-based networks	1 Mb/s	1 μs
Ethernet-based networks	10 Mb/s	100 ns

Quality of Service

Packet size and overhead

The data rate of a network must be considered together with the *packet size* and *overhead*.

Indeed, data is *encapsulated into packets*, with headers specifying the source and destination addresses of the packet, and often a checksum for detecting transmission errors.

All industrial networks have a *minimum packet size*, ranging from 47 bits for CAN to 84 bytes for Ethernet.

Moreover, a minimum *interframe time*, which is specific to each protocol, has to elapse between two successive frames to ensure that each packet can be distinguished individually.

The *overhead* is the set of bits used for *addressing* and *padding* the packet.

Quality of Service

Packet size and overhead

Figure: Typical transmitted frame.

Quality of Service

Transmission time

The *transmission time* t_{tx} is a quantity that can be measured over a network with a very high precision.

t_{tx} is given by the sum of two components:

- t_{fr} is the *frame time* corresponding to the time required to send the packet on the network;
- t_{pr} is the *propagation time* corresponding to the time needed to propagate the message between two devices.

Therefore, $t_{tx} = t_{fr} + t_{pr}$

Quality of Service

Frame time

The *frame time* t_{fr} obviously depends on the number of bits used in the packet encoding.
Depending on the adopted protocol, the frame may contain start/stop bits, error checking mechanisms, preamble, identifier, data, etc.
Usually, *padding* bits are also used to meet the minimum size number of a certain network.
The *frame time* is then given by

$$t_{fr} = N_{bits} t_{bit},$$

where N_{bits} is the overall number of bits contained in the frame.

Quality of Service

Frame time and data rate

The *frame time* and the packet encoding should be carefully considered along with the network *data rate*.
For example, to send one bit of data over a 500 kb/s CAN network, a 48 bit message is needed, hence a *frame time*:

$$t_{fr} = \frac{N_{bit}}{\text{data rate}} = \frac{48}{500\text{kb/s}} = 96\mu\text{s}.$$

To send the *same one bit* of data over 10 Mb/s Ethernet, an 84 byte message is needed (64 byte frame size plus 20 bytes for interframe separation), requiring a 67.2 μs.
Thus, even though the raw network data rate is 20 times faster for Ethernet, the frame time is only 30% lower than CAN.
This example shows that the network data rate *is only one factor* that must be considered when computing the effective data rate of a network.

Quality of Service

Propagation time

The *propagation time* t_{pr} is usually negligible since the *transmission speed* of a standard medium is typically in between $2 \cdot 10^8$ m/s and $3 \cdot 10^8$ m/s (depending on the medium).

For example, we usually have $t_{pr} = 67.2$ μs for a 2.5 km Ethernet cable or $t_{pr} = 1$ μs for a 100 m CAN bus.

t_{pr} is difficult to be characterised since it is dependent from the relative *distance between the devices*: it may be of some microseconds as the previous example or reach up to 280 ms for *geostationary satellite transmissions*.

Moreover, if Ethernet is adopted, the presence of switches, bridges, routers and hubs on the network generate a delay that is largely more relevant than the t_{pr}.

Quality of Service

Collision time

Another important factor to consider is the time spent for *packet collisions*, which usually entails *packet retransmission* and depends on *traffic*.
As already noticed, this is a function of the *Media Access Method* adopted in the MAC sublayer.

Outline

1. Trends in manufacturing networks
   - Parametrisation of Industrial Networks
   - Transmission times
   - Delays and jitters
2. Take home message

Quality of Service

Time delay and jitter

The *time delay* on a network is the total time between the data being available at the source node, e.g., sampled from the environment or computed by the controller, *and* the same data being available at the destination node, e.g., received and decoded.
Even though, many control techniques have been developed for systems with constant time delays, with *variability* the problems become worse.
The *jitter* is the variability of the delay.
Although time delay is an important factor to consider for control systems implemented over industrial networks, *it has not been well defined or studied* by standards organisations defining network protocols

Quality of Service

Time delay and jitter

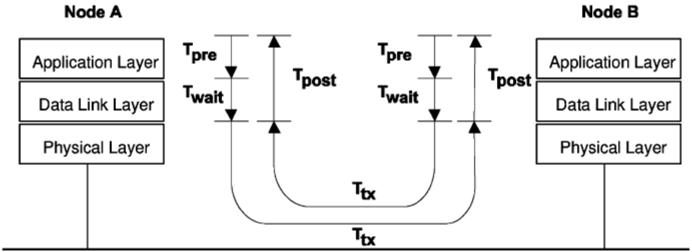


Figure: Timing diagram showing time spent sending a message from source node to destination node (James R. Moyne and Dawn M. Tilbury, "*The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data*", 2007). Therefore $t_{delay} = t_{pre} + t_{wait} + t_{tx} + t_{post}$.

Quality of Service

Pre and Post processing times

In the time delay

$$t_{delay} = t_{pre} + t_{wait} + t_{tx} + t_{post},$$

the *pre-processing* t_{pre} and *post-processing* t_{post} times are very difficult to be estimated, and also to be measured.
Moreover, they have usually a *large variability*.

Quality of Service

Wait time

The t_{wait} expresses the fact that *a message may spend time* waiting in the *queue* at the sender's buffer and could be blocked from transmitting by other messages on the network.
As recalled, this is related to the *traffic* and to the *MAC*.
The wait time variability is of course quite relevant for *Ethernet TCP/IP* networks, but it may be detrimental also for Fieldbuses due to the adopted *MAC* solutions.

Quality of Service

Wait time example: Strobe connection for a DeviceNet

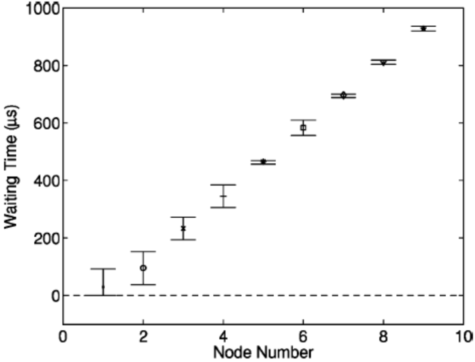


Figure: Nine identical devices on DeviceNet with strobed message connection: variability depends on the processing time (James R. Moyne and Dawn M. Tilbury, "*The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data*", 2007).

Quality of Service

Delay and jitter

Since the QoS is also a function of the *delay* and the *jitter* and since they are related to *unpredictable fluctuations*, their behaviour is quantified using a *stochastic description*.
The stochastic description is given in terms of the set of *measurements* that can be collected, as customary for every measurement system.

Quality of Service

Packet routing

When using large networks, especially if based on Ethernet, the packet may be *routed* in different directions.
Since the network is usually different depending on the choice of the link made, the *delay changes according to that choice*.
In *switched networks* the sources of these uncertainties are given by the presence of *switches* (or sometimes *hubs*).

Quality of Service

Packet routing

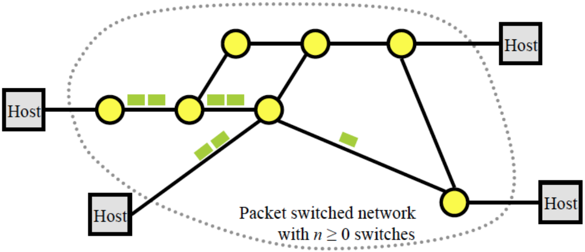


Figure: Packet delay is dependent from the particular route chosen (courtesy of University of Glasgow).

Quality of Service

Effect of packet routing

Let us make an example to understand the *effect of packet routing* over a channel with r routers and with *data rate* of m bs, where we assume $t_{pr} \approx 0$, i.e. $t_{tx} \approx t_{fr}$.

- Imagine that a node A transmit a message with p packets of N_{bit} each to a node B at time t_0;
- At time $t_0 + t_{tx}$, where $t_{tx} = N_{bit}/m$ s, the packet have entirely reached the first *router*;
- Hence, at time $t_0 + t_{tx}$ the packet can be retransmitted to the *second router*, and it will reach the *second router* at time $t_0 + 2t_{tx}$.
- It follows that at time $t_0 + (r + 1)t_{tx}$ the *first packet* reached node B. In other words the *end-to-end* delay is $\delta_{end-to-end} = (r + 1)t_{tx}$;
- As a consequence, the entire process lasts $p(r + 1)t_{tx}$.

Is there a better way to do it?

Quality of Service

Effect of packet routing

We can make use of *pipelining*: after the first packet reached the first router, we *can start with the transmission of the second packet*!
Therefore, the entire transmission process becomes $(r + 1 + p - 1)t_{tx} = (r + p)t_{tx}$ in place of $(pr + p)t_{tx}$.
WARNING! *Pipelining* has no effect on the *end-to-end* delay, since

$$\delta_{end-to-end} = (r + 1)t_{tx}$$

Quality of Service

Effect of packet routing

The effect of packet routing is even worse since *multiple links are connected together to the same switches* and, hence, *switches* need *message queues*.
Indeed, each *switch* has an *output queue* for each link: if a message finds the desired link *busy*, it should be stored in the *output queue*.
Of course this has a detrimental effect on *delays*.
But it can be even worse: since the amount of buffer space is *finite*, an arriving packet may find that the buffer is *completely full* with other packets waiting for transmission: whatever the policy, *one packet will be lost*.
Since the *traffic* has an immediate effect on the *message queue* status, it is evident that *the more the traffic, the more the probability of loosing packets* (either by *collision* or *message queues*).

Quality of Service

The golden rule

To clearly understand the connection between *traffic* and *message queues*, let us make this example.

- Suppose, for simplicity, we have a network where all the packets have length N_{bit} and the *data rate* is m bs;
- Suppose that the average *arrival rate* at the *message queue* in a switch is α s^{-1};
- Finally, suppose that the *message queue* as an *ideal infinite length*;
- It turns out that if $\alpha N_{bit}/m > 1$, then the queue reaches an *infinite length*!

$\alpha N_{bit}/m$ is the *traffic intensity*, which leads us to the *golden rule*: *design your system to have $\alpha N_{bit}/m < 1$*.

Quality of Service

Wait time: TCP/IP buffering

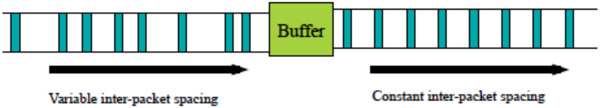


Figure: Large jitters ask for buffering along the line (courtesy of University of Glasgow).

To bound the *delays and jitters*, either *buffering* is applied along the line to reconstruct timing (e.g., *isochronous communication*) and/or a suitable *scheduling of messages* is adopted.

Quality of Service

Synchronisation

We noticed that the distance among the sender and receiver are responsible of wrong *interarrival-time* of packets.
This is of particular relevance for *isochronous* networks.
According to the *Oxford Dictionary*:

Definition

An *isochronous network* is a high-data rate network for which the latency can be easily predicted. Such networks are used for applications such as video on demand.

Strictly speaking, in isochronous networks the correctness of the data depends on *both* the *data content* and the *time* in which it has arrived (the same concept of RT).
Example: tracking a target in a distributed surveillance system.

Quality of Service

Synchronisation

The main problem is that the internal clock of each device is *free-running* and usually *unsynchronised*.
Hence, it results in a steady increase or decrease in the inter-packet spacing observed at the receiver.
The solution is to adopt specifically conceived *synchronisation protocols* to synchronise the network clocks.

Quality of Service

Other measures

Other measures of QoS can be adopted:

- *Reliability of data transmission*: Data can be corrupted during transmission. To overcome this problem, some network adopt a *handshake* mechanism by sending *acknowledgment* packets (which introduce overhead);
- *Security*: Used when data are vulnerable to malicious attacks or viruses. Fieldbuses prevent this problem by detaching the internal network from the outer network. In this case, the secure connection are used to *prevent misuses* of process data rather than preventing malicious attacks. More on this issue at the end of the slides.

Security in industrial networks

Security is not safety!

Security is not *safety*!

Definition

Safety is the state of being *safe*, i.e., the condition of being protected against physical failure, damage, error, accidents, harm, or any other event that could be considered non-desirable.

Definition

Security is the degree of robustness/resistance to or protection from harm.

For networks, the term *security* refers to the policies adopted to prevent and monitor authorised access, misuse, modification, or malicious usage of network-accessible resources.

Quality of Service

QoS vs QoC

The QoS offered by the network is not the only parameter to consider in a *control network*.
Indeed, for control applications, the *Quality of Control* (QoC) measure is of major importance.
The QoC can be measured using standard measures in control theory, e.g., *steady state value*, *settling time*, *rise time*, *mean square stability*, etc.
In practice, the QoC *is a function* of the QoS offered by the network.

Quality of Service

QoS vs QoC

Another important player for QoC on control networks is related to the *digital controller* implementation: the *continuous time* controller should be sampled.

The digital controller is a function of the *sampling time* adopted.

Ideally, the *shorter* is the sampling time, the *closer* is the continuous time behaviour of the plant.

However, this represents a problem for *networked* and *distributed* control systems, since the number of data to be transmitted over the network increases as well, hence increasing latencies, delays, traffic, etc.

Usually the choice of the sampling is *traded* between this two contrasting goals.

Quality of Service

QoS vs QoC

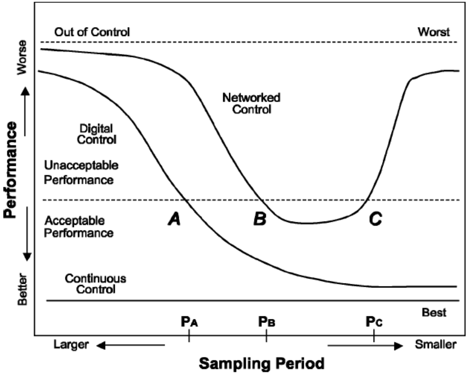


Figure: Performance comparison of continuous control, digital control, and networked control, as a function of sampling frequency (James R. Moyne and Dawn M. Tilbury, "*The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data*").

Control Networks

Architecture example

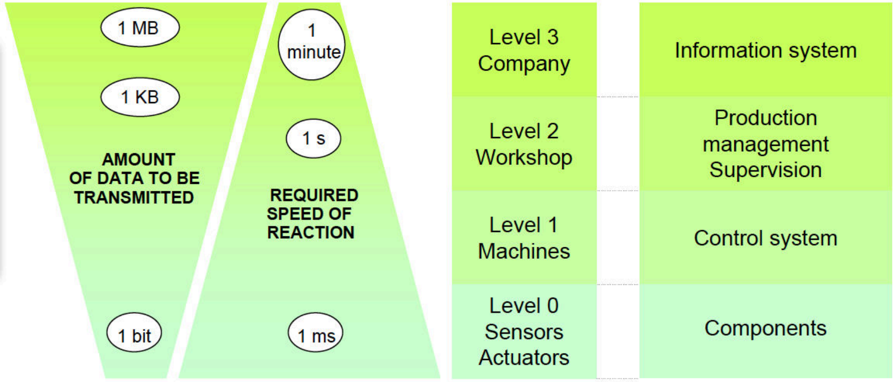


Figure: Example of required QoS (Courtesy of Schneider Electric).

Outline

1. Trends in manufacturing networks
   - Parametrisation of Industrial Networks
   - Transmission times
   - Delays and jitters
2. Take home message

Control networks

- Networks can be classified using their *Quality of Service*.
- The QoS generally comprises: *throughput*, *delays*, *jitters*, *reliability* and *security*.
- Roughly speaking, the *packet delays* are a function of the medium and the *MAC* adopted.
- Jitters are instead a function of the *MAC* and of the *queues*, hence of the network *traffic*.
- The QoS has an impact on the *Quality of Control* (QoC) that can be achieved.
- *Industrial networks* are usually an integration of different solutions among Fieldbuses and standard Ethernet-based solutions.