Instituto Superior de Engenharia do Porto

P.PORTO

Communication Technologies for Critical Systems

Communication Protocols: Ehternet-based protocols

Mestrado em Engenharia de Sistemas Computacionais Críticos

Ethernet

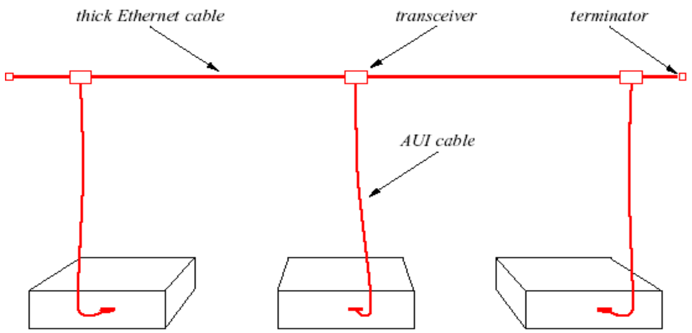
- Most successful local area networking technology of the last 20 years.
- Developed in the mid-1970s by researchers at the Xerox Palo Alto Research Centers (PARC).
- Uses CSMA/CD technology
  - Carrier Sense Multiple Access with Collision Detection
  - A set of nodes send and receive frames over a shared link
  - Carrier sense means that all nodes can distinguish between an idle and a busy link
  - Collision detection means that a node listens as it transmits and can therefore detect when a frame it is transmitting has collided with a frame transmitted by another node

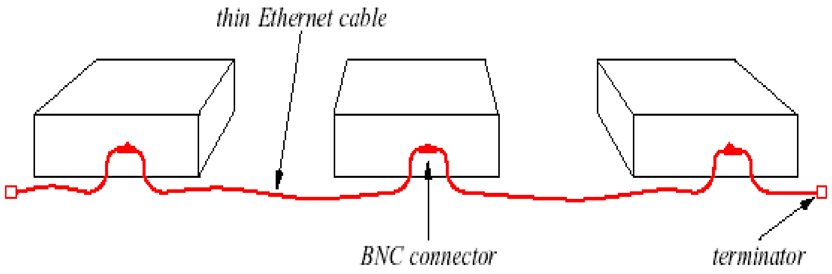
Ethernet

- Uses ALOHA (packet radio network) as the root protocol
  - Developed at the University of Hawaii to support communication across the Hawaiian Islands.
  - For ALOHA the medium was atmosphere, for Ethernet the medium is a coax cable.
- DEC and Intel joined Xerox to define a 10-Mbps Ethernet standard in 1978.
- This standard formed the basis for IEEE standard 802.3
- More recently 802.3 has been extended to include a 100-Mbps version called Fast Ethernet and a 1000-Mbps version called Gigabit Ethernet.

Ethernet Generations

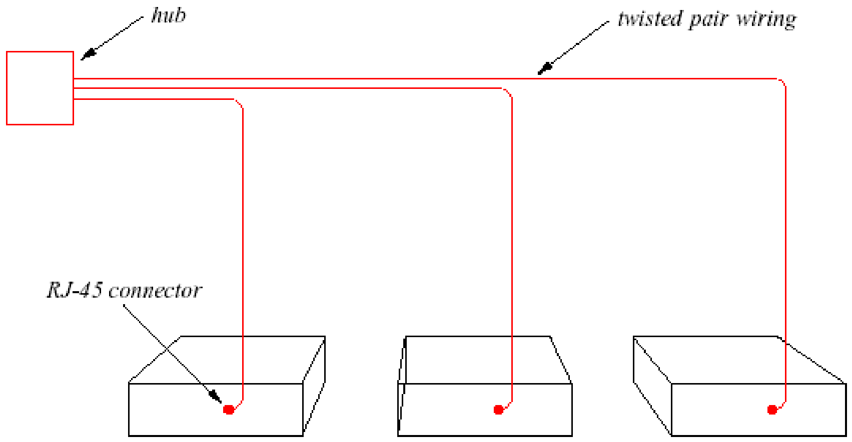
- **Original Ethernet:**
  - Coaxial cable (10Base5)
  - Thicknet.





- **Next Generation:**
  - Thin coax cable (10Base2)
  - Thinnet.

- **Modern Ethernet:**
  - Twisted pair ethernet (10BaseT)
  - Uses hub: physical star but logical bus.



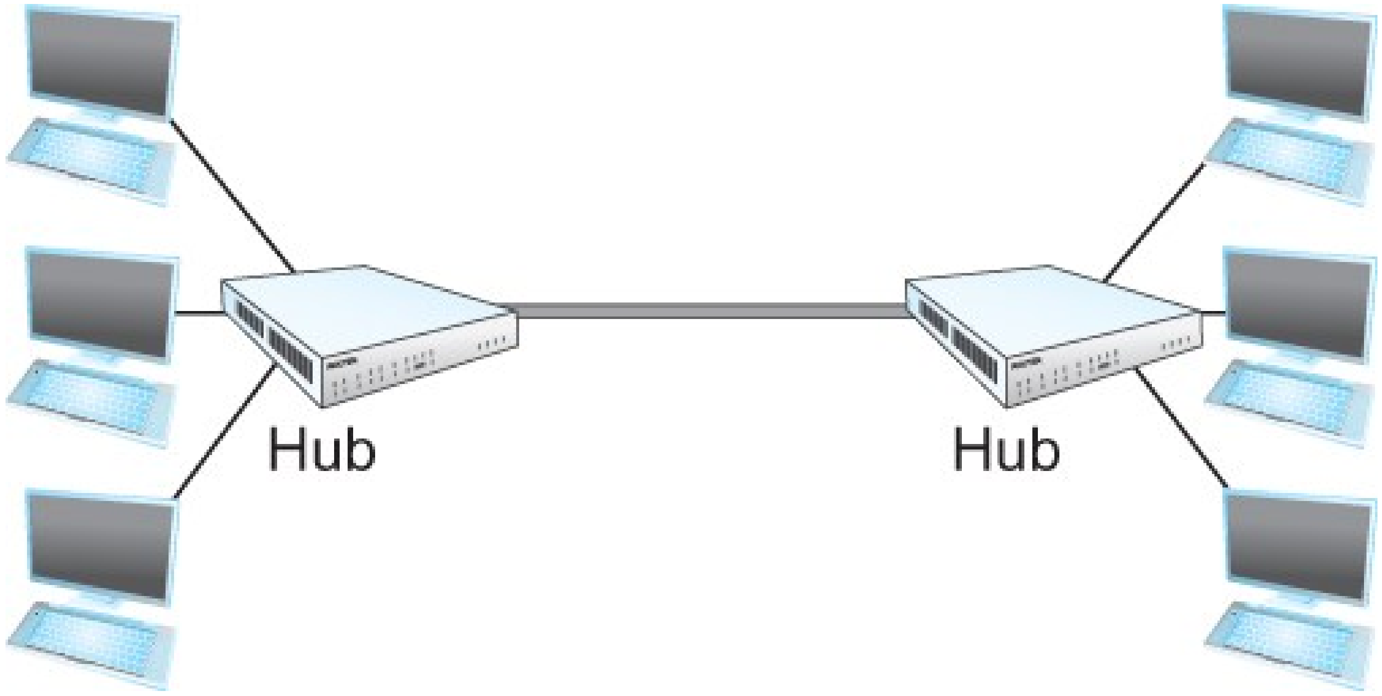
Ethernet

- New Technologies in Ethernet
  - Instead of using coax cable, an Ethernet can be constructed from a thinner cable known as 10Base2 (the original was 10Base5)
    - 10 means the network operates at 10 Mbps
    - Base means the cable is used in a baseband system
    - 2 means that a given segment can be no longer than 200 m

Ethernet

- New Technologies in Ethernet
  - Another cable technology is 10BaseT
    - T stands for twisted pair
    - Limited to 100 m in length
  - With 10BaseT, the common configuration is to have several point-to-point segments coming out of a multiway repeater, called *Hub*

Ethernet

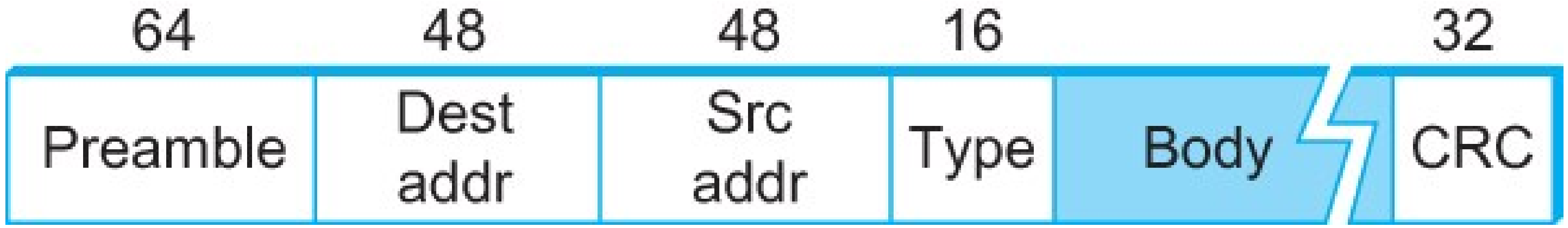


Ethernet Hub

Access Protocol for Ethernet

- The algorithm is commonly called Ethernet's Media Access Control (MAC).
  - It is implemented in Hardware on the network adaptor.
- Frame format
  - Preamble (64bit): allows the receiver to synchronize with the signal (sequence of alternating 0s and 1s).
  - Host and Destination Address (48 bit each).
  - Packet type (16 bit): acts as demux key to identify the higher-level protocol.
  - Data (up to 1500 bytes)
    - Minimally a frame must contain at least 46 bytes of data.
    - Frame must be long enough to detect collision.
  - CRC (32bit)

Ethernet Frame



Ethernet Frame Format

Ethernet Addresses

- Each host on an Ethernet (in fact, every Ethernet host in the world) has a unique Ethernet Address – its **MAC address**
- Ethernet addresses are typically printed in a human readable format
  - As a sequence of six numbers separated by colons.
  - Each number corresponds to 1 byte of the 6 byte address and is given by a pair of hexadecimal digits, one for each of the 4-bit nibbles in the byte
  - Leading 0s are dropped.
  - For example, **8:0:2b:e4:b1:2** is
    - 00001000 00000000 00101011 11100100 10110001 00000010
- To ensure that every adaptor gets a **unique address**, each manufacturer of Ethernet devices is allocated a different prefix that must be prepended to the address on every adaptor they build
    - AMD has been assigned the 24bit prefix 8:0:20

Ethernet Addresses

- Each frame transmitted on an Ethernet is received by every adaptor connected to that Ethernet.
- Each adaptor recognizes those frames addressed to its address and passes only those frames on to the host.
- In addition, to *unicast* address, an Ethernet address consisting of all 1s is treated as a *broadcast* address.
  - All adaptors pass frames addressed to the *broadcast* address up to the host.
- Similarly, an address that has the first bit set to 1 but is not the *broadcast* address is called a *multicast* address.
  - A given host can program its adaptor to accept some set of *multicast* addresses.

Ethernet MAC Algorithm

- When the adaptor has a frame to send and the line is idle, it transmits the frame immediately.
  - The upper bound of 1500 bytes in the message means that the adaptor can occupy the line for a maximum amount of time.
- When the adaptor has a frame to send and the line is busy, it waits for the line to go idle and then transmits immediately.
- The Ethernet is said to be 1-persistent protocol because an adaptor with a frame to send transmits with probability 1 whenever a busy line goes idle.

Ethernet Transmitter Algorithm

- Since there is no centralized control it is possible for two (or more) adaptors to begin transmitting at the same time,
  - Either because both found the line to be idle,
  - Or, both had been waiting for a busy line to become idle.

- When this happens, the two (or more) frames are said to be *collide* on the network.

Ethernet Transmitter Algorithm

- Since Ethernet supports collision detection (CD), each sender can determine that a collision is in progress.
- At the moment an adaptor detects that its frame is colliding with another,
  - it first makes sure to transmit a 32-bit jamming sequence
  - And, then stops transmission
  - Thus, a transmitter will minimally send 96 bits in the case of collision
    - 64-bit preamble + 32-bit jamming sequence

Ethernet Transmitter Algorithm

- One way that an adaptor will send only 96 bit (called a *runt frame*) is if the two hosts are close to each other.
- Had they been farther apart,
  - They would have had to transmit longer, and thus send more bits, before detecting the collision.

Ethernet Transmitter Algorithm

- The worst-case scenario happens when the two hosts are at opposite ends of the Ethernet.

- To know for sure that the frame it has just sent did not collide with another frame, the transmitter may need to send as many as 512 bits.
  - Every Ethernet frame must be at least 512 bits (64 bytes) long.
    - 14 bytes of header + 46 bytes of data + 4 bytes of CRC

Ethernet Transmitter Algorithm

- Why 512 bits?
  - Why is its length limited to 2500 m?

- The farther apart two nodes are, the longer it takes for a frame sent by one to reach the other, and the network is vulnerable to collision during this time

Ethernet Transmitter Algorithm

- A begins transmitting a frame at time *t*
- *d* denotes the link latency
- The first bit of A's frame arrives at B at time $t + d$
- Suppose an instant before host A's frame arrives, host B begins to transmit its own frame
- B's frame will immediately collide with A's frame and this collision will be detected by host B
- Host B will send the 32-bit jamming sequence
- Host A will not know that the collision occurred until B's frame reaches it, which will happen at $t + 2 * d$
- Host A must continue to transmit until this time in order to detect the collision
  - Host A must transmit for $2 * d$ to be sure that it detects all possible collisions

Ethernet Transmitter Algorithm

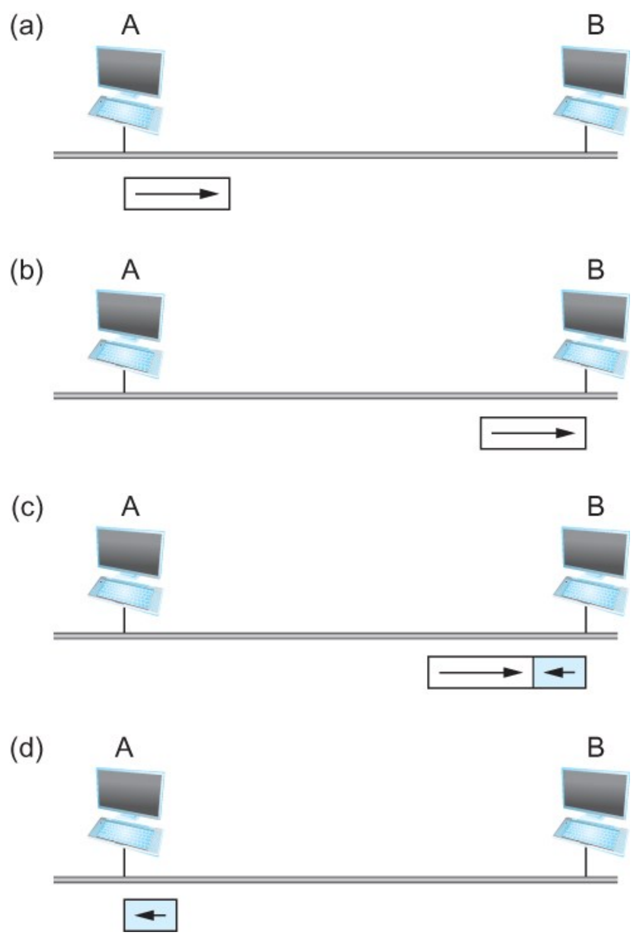
Worst-case scenario:

(a) A sends a frame at time *t;*

(b) A's frame arrives at B at time *t + d;*

(c) B begins transmitting at time t + d and collides with A's frame;

(d) B's runt (32-bit) frame arrives at A at time *t* + *2D.*



Ethernet Transmitter Algorithm

- Consider that a maximally configured Ethernet is 2500 m long, and there may be up to four repeaters between any two hosts, the round trip delay has been determined to be 51.2 μs
  - Which on 10 Mbps Ethernet corresponds to 512 bits
- The other way to look at this situation,
  - We need to limit the Ethernet's maximum latency to a fairly small value (51.2 μs) for the access algorithm to work
    - Hence the maximum length for the Ethernet is on the order of 2500 m.

Ethernet Transmitter Algorithm

- Once an adaptor has detected a collision, and stopped its transmission, it waits a certain amount of time and tries again.
- Each time the adaptor tries to transmit but fails, it doubles the amount of time it waits before trying again.
- This strategy of doubling the delay interval between each retransmission attempt is known as *Exponential Backoff*.

Ethernet Transmitter Algorithm

- The adaptor first delays either 0 or 51.2 μs, selected at random.
- If this effort fails, it then waits 0, 51.2, 102.4, 153.6 μs (selected randomly) before trying again;
  - This is k * 51.2 for k = 0, 1, 2, 3
- After the third collision, it waits k * 51.2 for $k = 0 \ldots 2^3 - 1$ (again selected at random).
- In general, the algorithm randomly selects a k between 0 and $2^n - 1$ and waits for k * 51.2 μs, where n is the number of collisions experienced so far.

Bibliography

- https://inst.eecs.berkeley.edu/~ee290f/sp04/kc_ethernet.ppt
- Lecture material taken from “Computer Networks A Systems Approach”, Third Ed.,Peterson and Davie, Morgan Kaufmann, 2003

Real-time Ethernet

Real-time Ethernet: Motivation

- Since the 80's several dedicated network protocols (fieldbuses) have been proposed
  - CAN, DeviceNet, Profibus, SERCOS, LonWorks, WorldFIP, P-Net, Interbus, AS-Interface, MIL-STD-1553 and countless others
- However, “traditional” fieldbuses:
  - Have limited performance (compared with Ethernet)
  - Integration with the Internet difficult
  - Proprietary (dependency of a single supplier)
  - Costly in terms of installation and maintenance (niche market)
- **So conventional fielbuses are becaming ... OBSOLETE!**

Motivation

- Ethernet is currently on the way of becoming the prevalent technology in automation networks
- Standardized, mature field-proven technology
- **Many technology suppliers**
- **Well-known technology**: widely available expertise and tools
- Large bandwidth, with clear path for future expansion, enough to satisfy the forecast needs
- Good Internet integration
- **Appealing price/performance relationship**
- Several extensions for real-time operation
- Operation at the fieldbus level enabled

Why Real-time Ethernet

- Despite having many convenient features, “standard” Ethernet fails in key points:
  - Poor determinism
  - Poor scheduling capabilities
  - Lack of support to specific traffic types
    - E.g. time-triggered traffic
- **Consequently**, real-time extensions to Ethernet (RTE) have been developed to address these issues

PI PROFIBUS • PROFINET

TTEthernet

ETHERNET POWERLINK

EtherCAT – The real-time Ethernet fieldbus

HaRTES Hard Real-Time Ethernet Switching

EtherNet/IP™

SERCOS interface

Different approach to achieve determinism

- COTS Ethernet switches with planning
  - Ethernet/IP
  - Control the network load, low utilization and high jitter
- Traffic shaping in the end nodes
  - Linux Traffic Control, Traffic shaping
  - Prevents memory overflows but still suffers from high jitter due to the FIFO queues and eventually non-deterministic MAC
- Master-slave
  - EtherCAT, Ethernet Powerlink, FTT-SE
  - Not compatible with ordinary Ethernet nodes, limitations to the traffic types supported
  - High overhead (for “pure” master-slave)

Different approach to achieve determinism

- TT-oriented
  - PROFINET-IRT, TTEthernet
  - Tuned for isochronous traffic
  - High overhead (TDMA scheme – guard time)
  - Complex pre-planning of the whole network, requires specific hardware
- ET-oriented
  - AFDX, AV-Bridges
  - Tuned for asynchronous traffic, requires specific hardware
  - Isochronous traffic suffers high jitter.
- HaRTES
  - Supports both TT and ET paradigms, requires specific hardware

Focus

- Profinet IRT
- Time Sensitive Networks (TSN)

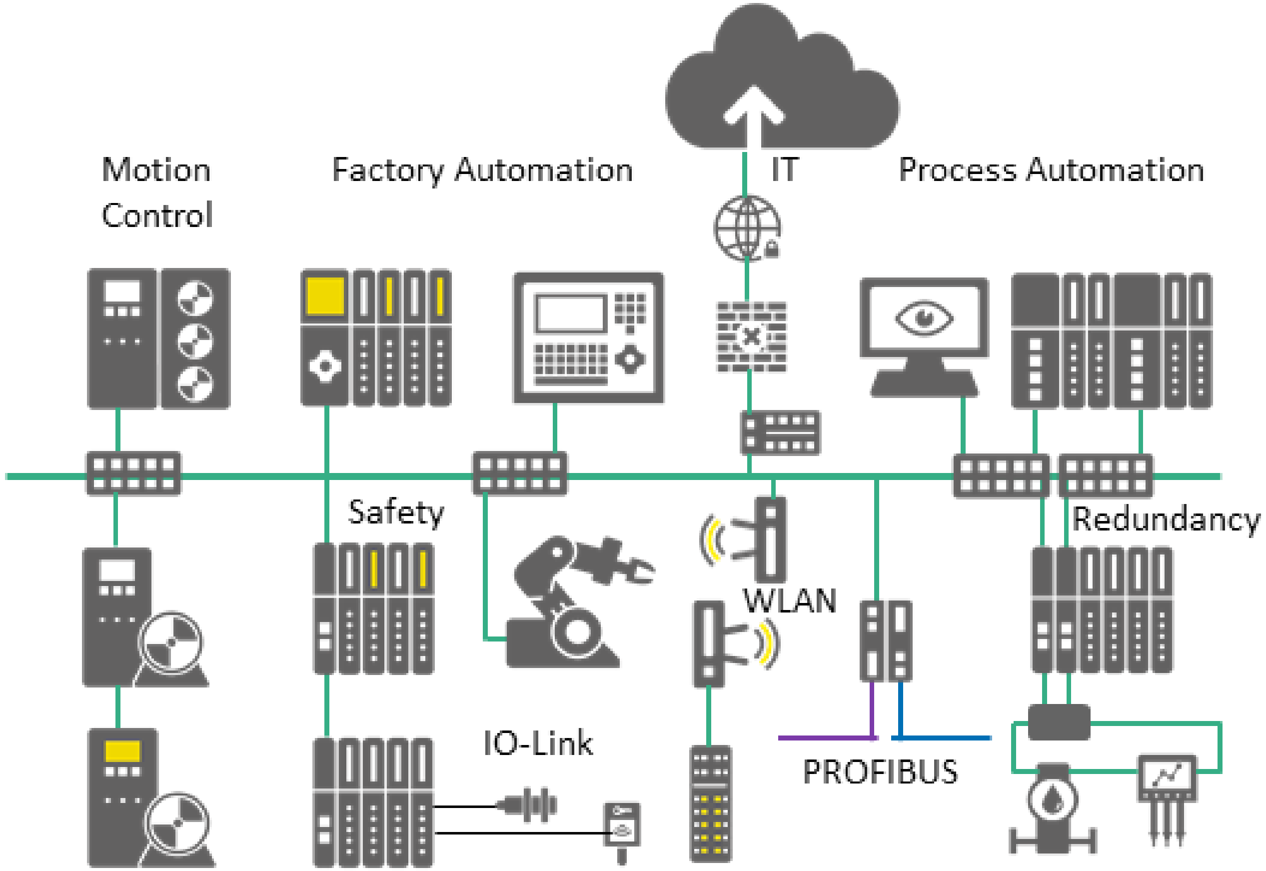
PROFINET

Introduction

- PROFINET - Industrial Ethernet Standard developed by PROFIBUS International
- Non-Real-time communication
  - Process automation
  - Standard TCP/IP UDP/IP stacks
- Real-time communication
  - Factory automation
  - Cycle times >= 5ms
- Isochronous-Real-Time (IRT) communication
  - High performance factory automation applications, like motion control
  - Cycle times <=1ms; jitter < 1µs
  - Requires special hardware

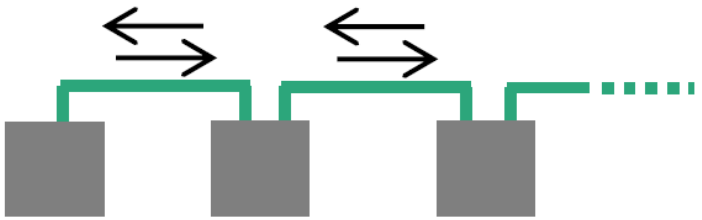
PROFINET

- A networking solution of production and process automation
- Supports applications with functional safety requirements and the entire spectrum of drive technology to isochronous motion control applications.
- Supports application profiles allowing optimal usage of PROFINET in all areas of automation engineering



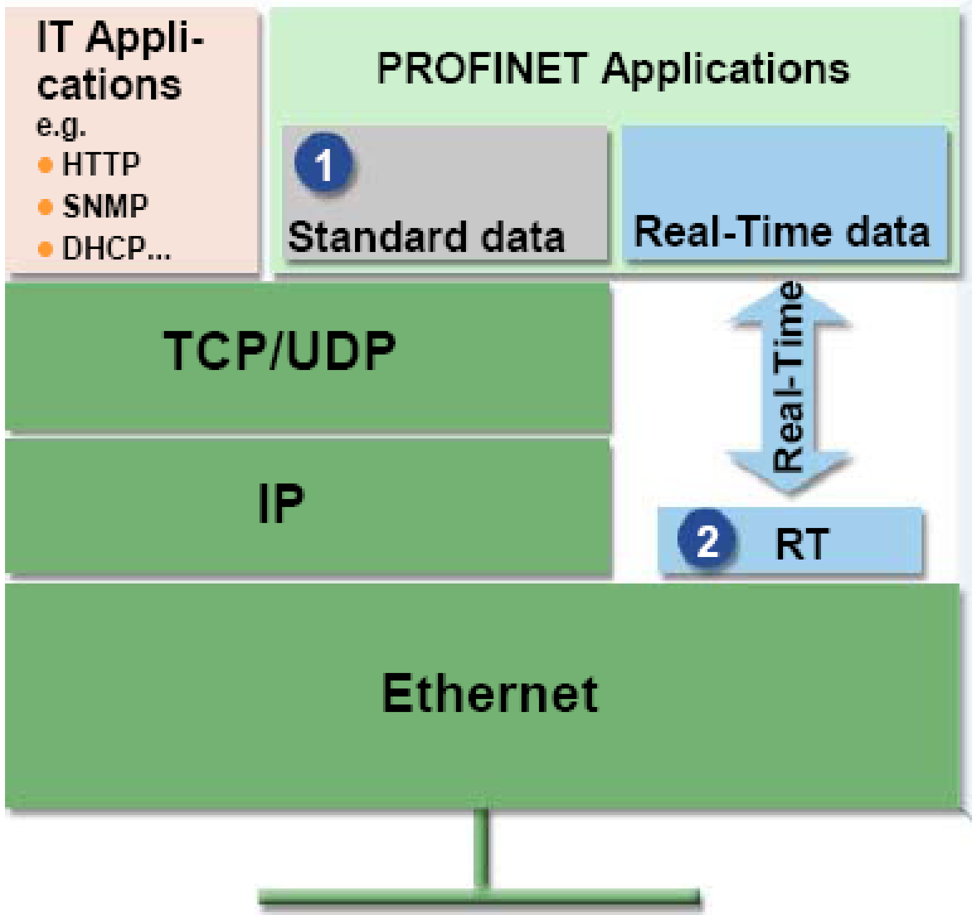
PROFINET: main charateristics

- Standard Ethernet IEEE802.3
- Switching technology IEEE802.1Q
- Wireless LAN IEEE802.11
- Flexible network topologies
- Physical Port to Port communication, Copper 100m; FO up to 80 km (APL)
- Two wire technology (APL)
- Switch integration into the devices
- PROFINET and Standard Ethernet devices mixed in one network



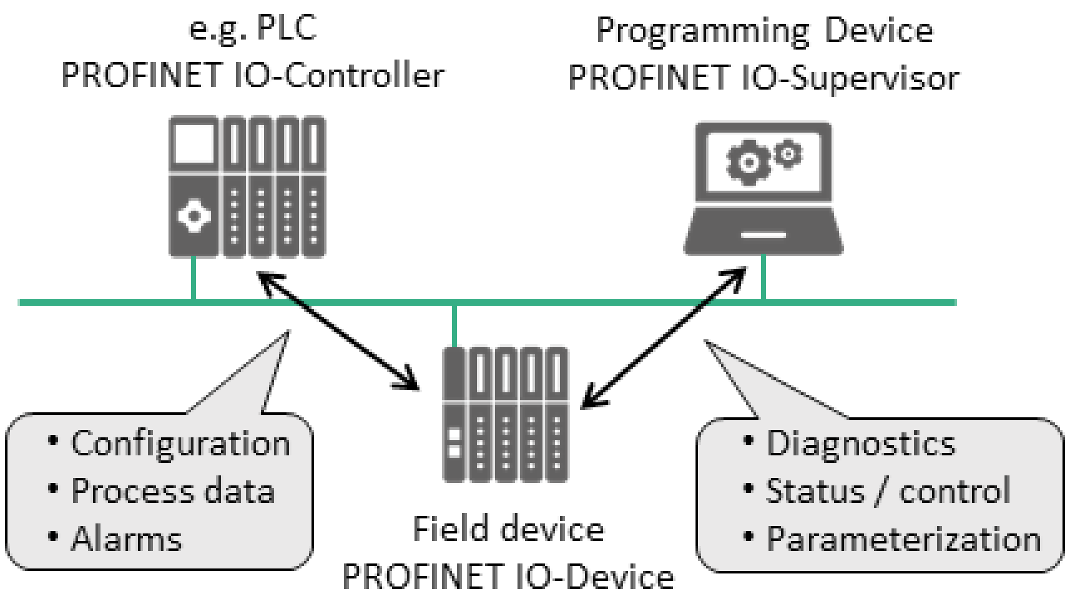
Switches connected into a line

PROFINET: communication layers



System model

- PROFINET follows the provider/consumer model for data exchange. This means that both the IO controller and IO device spontaneously send cyclic data independently.

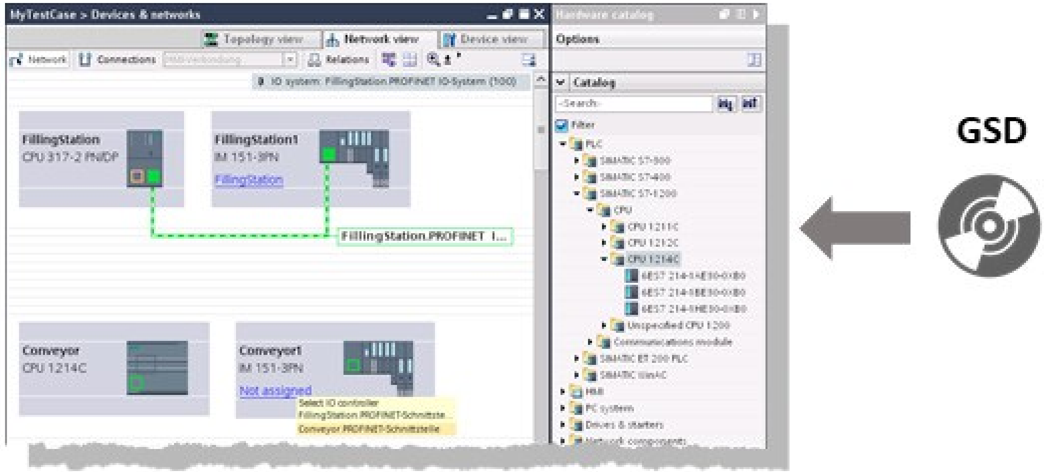


Types of devices

- IO controller:
  - Typically, a Programmable Logic Controller (PLC) in which the automation program runs. Provides output data to the configured IO devices in its role as provider and is the consumer of input data.
- IO device:
  - A distributed IO field device connected to one or more IO controllers via PROFINET. It is the provider of input data and the consumer of output data from the IO controller.
- IO supervisor:
  - This can be a programming device (PG), personal computer (PC) or human machine interface (HMI) device for commissioning or diagnostic purposes

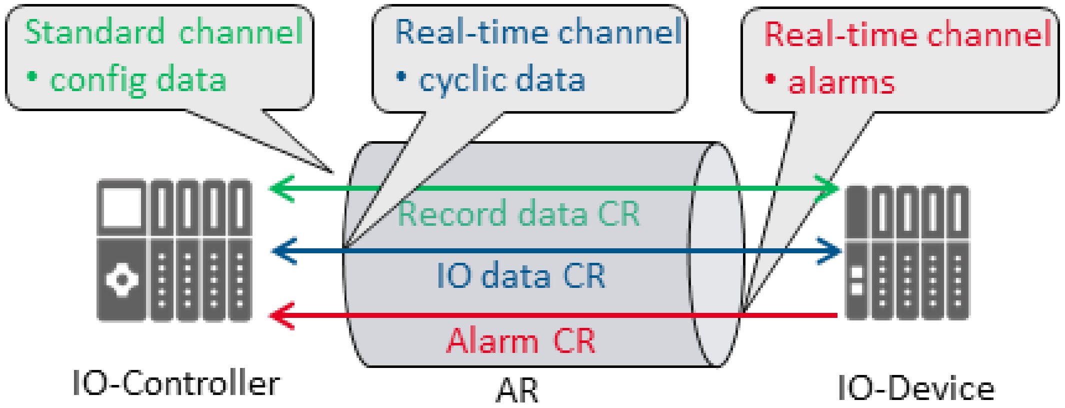
Profiles

- GSD files (General Station Description)
  - Describes the properties and functions of the PROFINET field device, including its modules and submodules
  - contains all data relevant for engineering as well as for data exchange with the field device.
  - Every manufacturer of a PROFINET field device must create an associated GSD file. It will be checked as part of the certification test

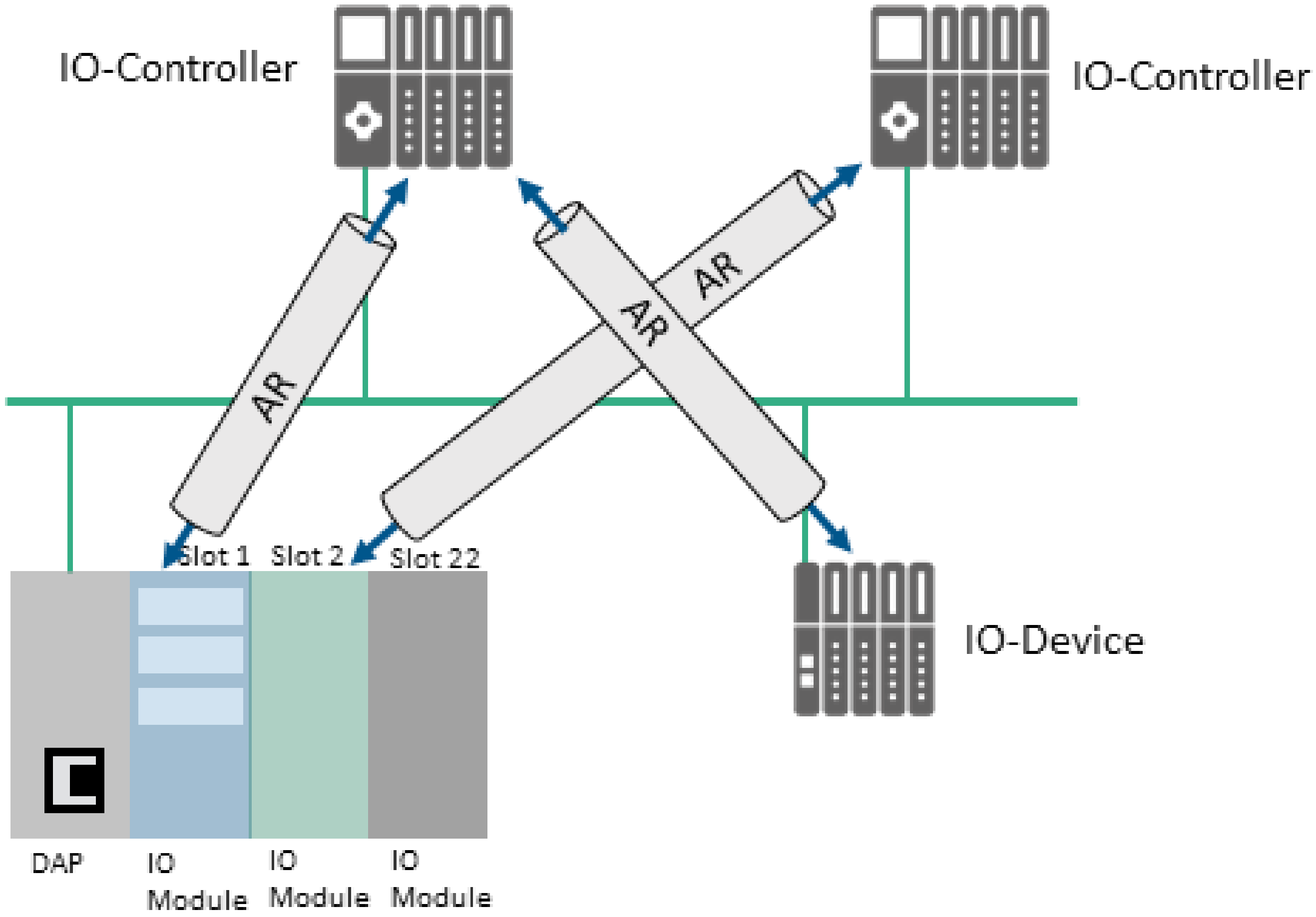


Communication Relations

- All data exchange is embedded into an AR (Application Relation). Within the AR, CRs (Communication Relations) specify the data explicitly.
- All data for device modeling, including the general communication parameters, are downloaded to the IO device. An IO device can have multiple ARs established from different IO controllers, for example, for shared devices.



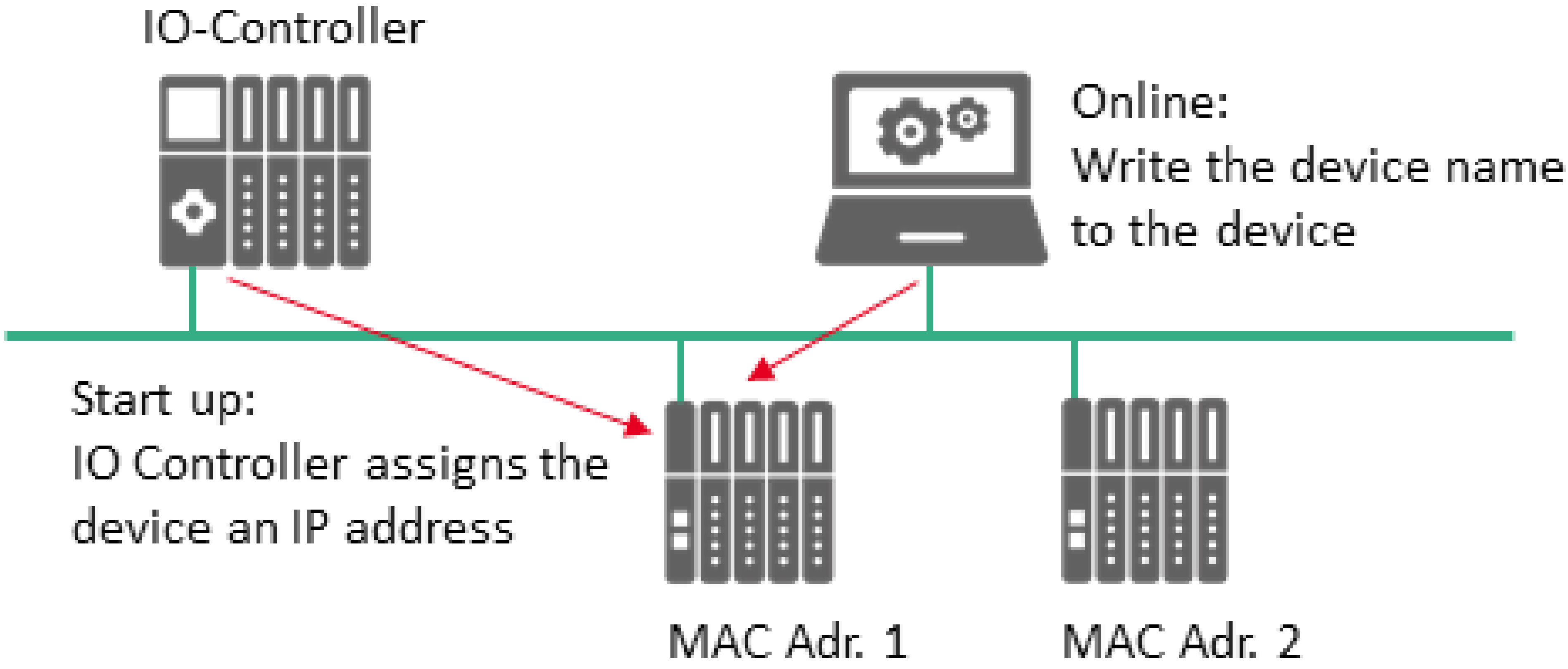
Communication relations



PROFINET addressing

- In a PROFINET system, each field device receives a symbolic name which uniquely identifies the field device within the IO system using the DCP protocol (Discovery and basic Configuration Protocol).
- Device name is assigned to an individual IO device and thus to its MAC address by an engineering tool during commissioning (device initialization).
- IP addresses are assigned via DHCP (Dynamic Host Configuration Protocol) or using manufacturer-specific mechanisms.

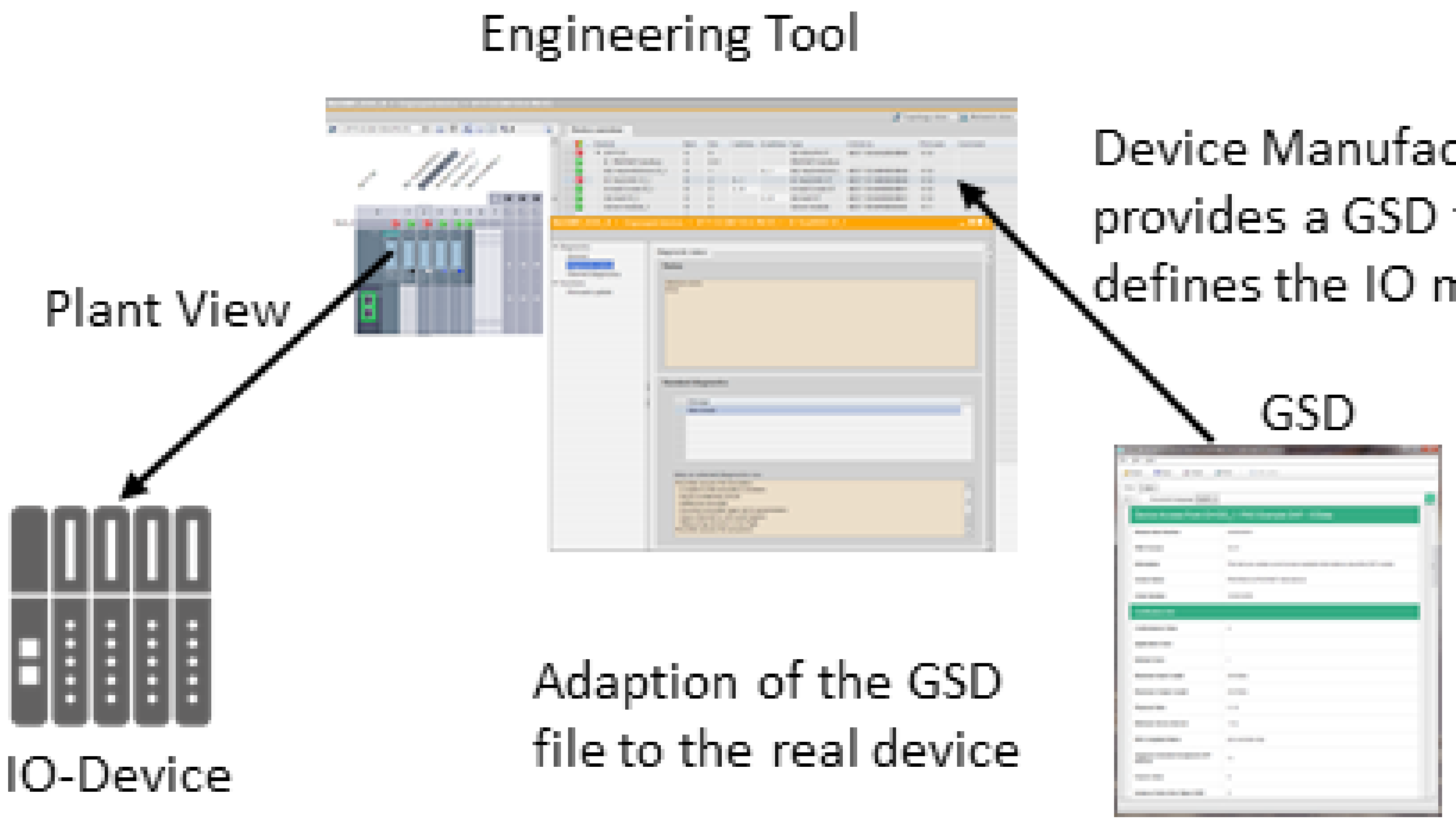
PROFINET addressing



Engineering/comissioning of an IO System

- IO controllers are supported by an engineering tool for configuring a PROFINET system. Configuring:
  - Cyclic data exchanges
  - Communication relations
  - population of the slots and subslots with modules and submodules
  - Other properties of the devices and modules/submodules can be specified using parameters
  - ...
- During commissioning, the configuration of the PROFINET IO system is downloaded to the IO controller. This means that the IO controller has all the information for addressing and data exchange with the IO devices

Engineering/comissioning of an IO System



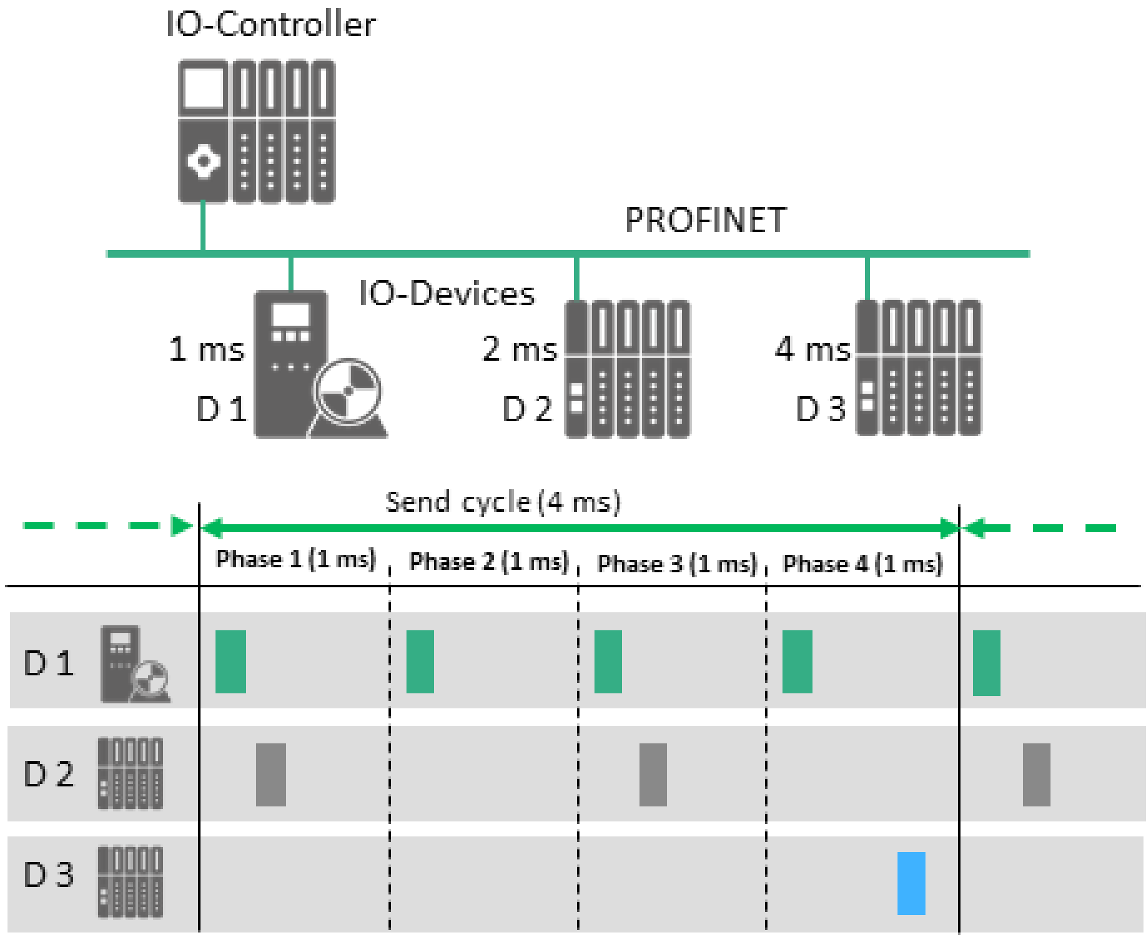
Basic functions

- The cyclical exchange of IO data with real-time properties.
- Acyclic data traffic for reading and writing need-based data (parameters and diagnostic data), including the identification and maintenance function (I&M) for reading out device information.
- A flexible alarm model for signaling device and network errors with three alarm levels (maintenance requirement, urgent maintenance requirement and diagnostics).

Cyclic Data Exchange

- Transmitted via the "IO data CR“ unacknowledged as real-time data between the provider and consumer in a parameterizable time frame.
- The update time can be specified individually for connections to the individual devices and are thus adapted to the requirements of the application.
- Different update times can be selected for the input and output data within the range of 250 µs to 512 ms.

Cyclic Data Exchange



Acyclic Data Exchange

- Acyclic data exchange is supported by the "record data CR“ and can be used for parameter configuration and other configuration of the IO devices or reading out status information.
  - Implemented over read / write services based on UDP
  - Also by manufacturer specific, web server, OPC UA,...

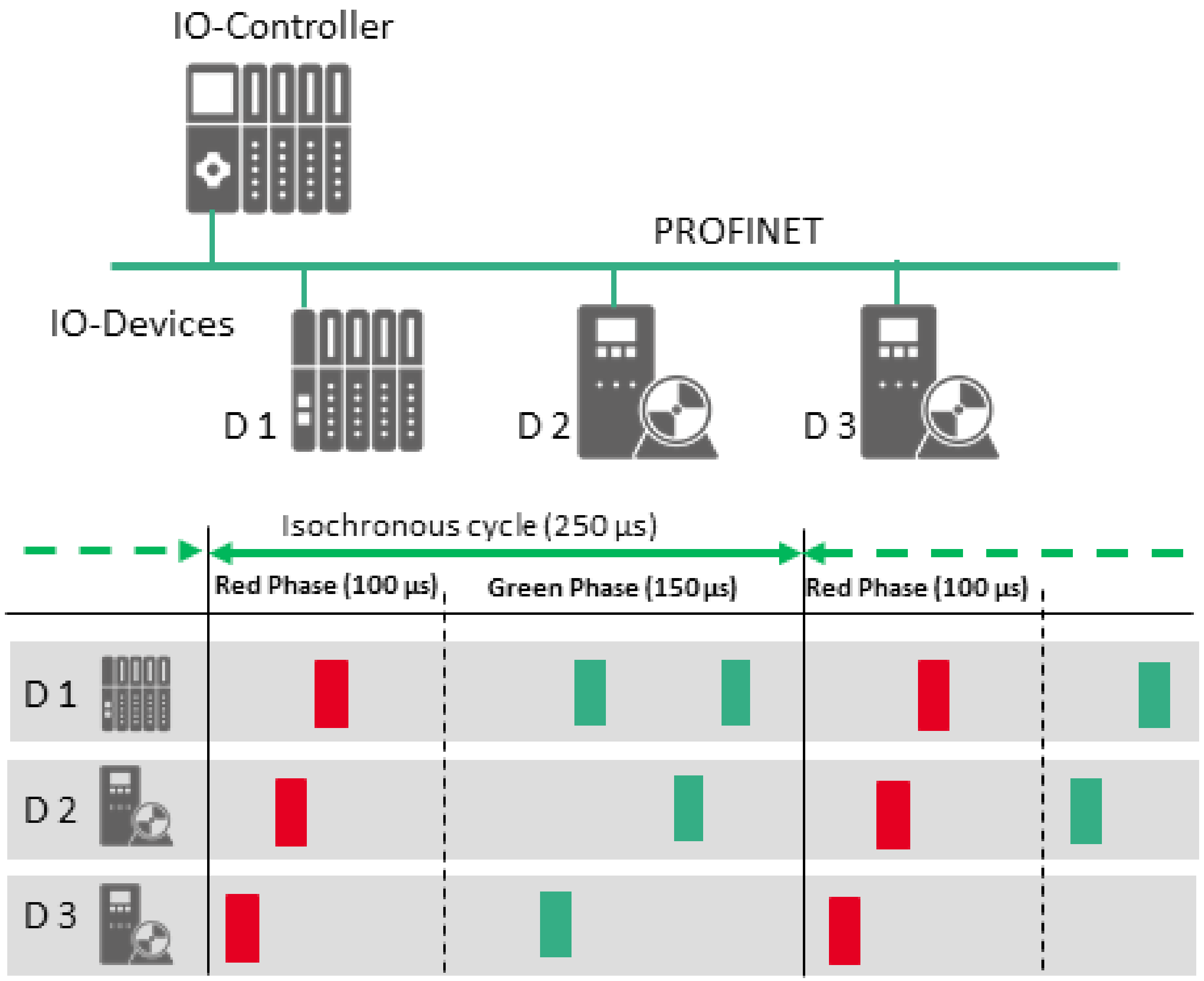
System data records

- **Diagnostic information** about the network and the devices can be read out by the user from any device at any time
- **Identification and Maintenance** (I&M) data supports unique identification of the devices, modules and submodules and their versions. This identification information is an important basis for maintaining the system and for asset management.
  - Information specified in the I&M data structures.
    - IM0: information about the hardware and firmware versions of the field device
    - IM1-3: system-dependent information, such as the installation location and date, and are created during configuration and written to the device
    - IM4 is used for a signature with PROFIsafe
    - IM5 provides information about the communication module of a field device.

Synchronous Real-Time

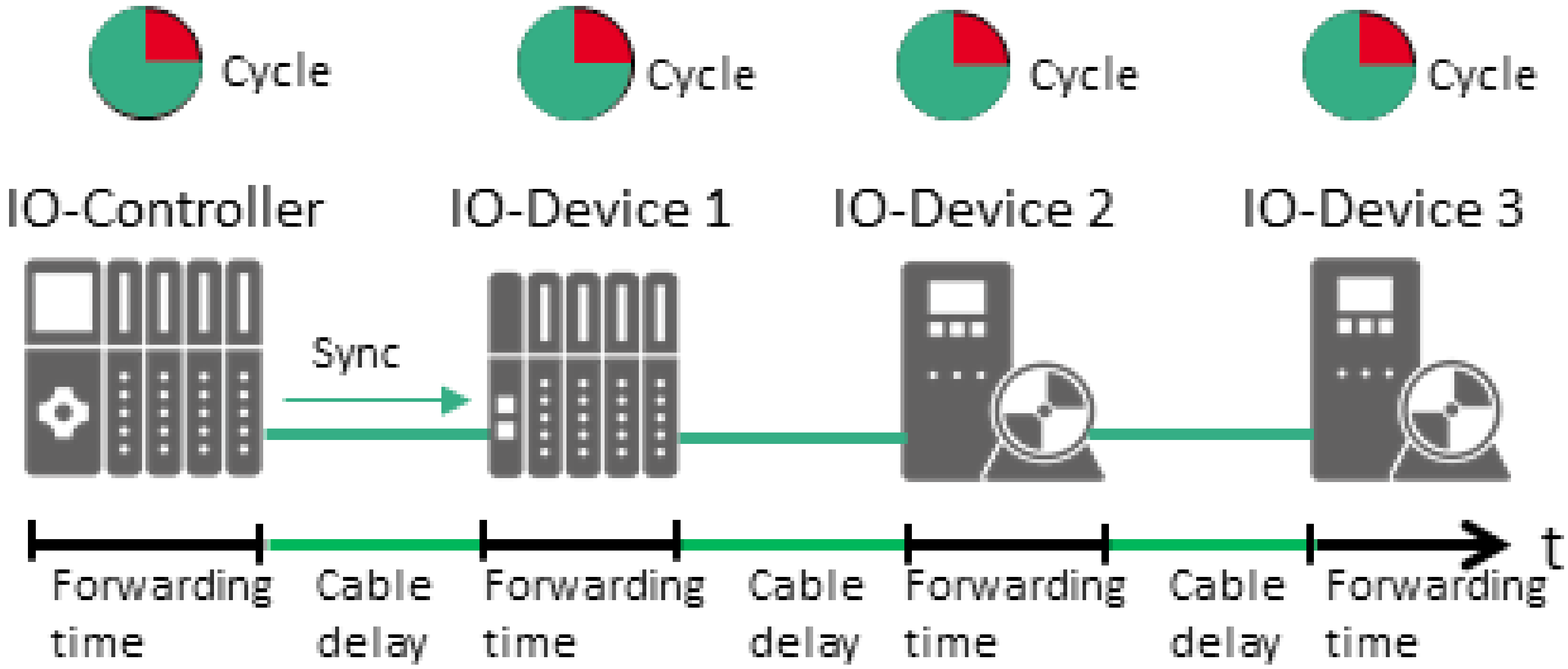
- Conformance Class C includes all necessary network-wide synchronization functions for applications with the most stringent requirements for deterministic behavior.
  - jitter of less than 1 microsecond (1 µs).
  - The cyclic data packets are transferred as synchronized packets during a reserved bandwidth (red phase)
  - All other packets, such as packets for diagnostics or TCP/IP, share the remaining Ethernet bandwidth (green phase).
  - Minimum update rate is defined at 250 µs, minimum of 31.25 µs, depending on the hardware used.

Synchronous Real-Time



Cycle Synchronization

- For the bus cycles to run synchronously (at the same time) with a maximum deviation of 1 μs, all devices involved in synchronous communication must have a common clock.
- A clock master uses synchronization frames to synchronize all local clock pulse generators of devices within a clock system (IRT domain) to the same clock.


Cycle
Cycle
Cycle
Cycle
IO-Controller
IO-Device 1
IO-Device 2
IO-Device 3
Sync
t
Forwarding time
Cable delay
Forwarding time
Cable delay
Forwarding time
Cable delay
Forwarding time

Cycle Synchronization

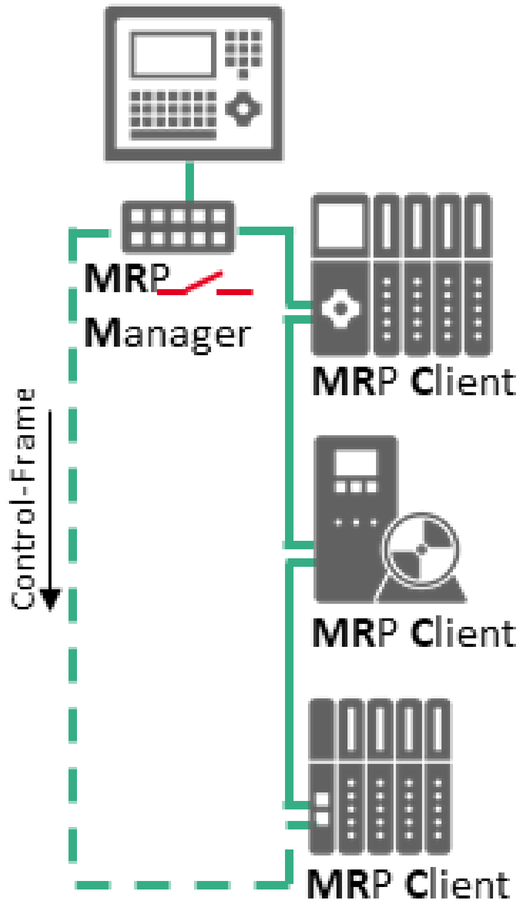
- To achieve the desired accuracy for the synchronization and synchronous operation, the **cable delay** on each connecting cable must be determined via measurement using defined Ethernet frames.
- Together with the **forwarding times** of the device or switches, all time ratios in the IRT system are known and can be figured into the synchronization.
  - If the IO controller (clock master) sends out a synchronization frame with the time of the clock master which reaches IO device 3 after 10 µs, IO device 3 experiences the frame having been underway for 10 µs and that the conveyed time of the clock master is 10 µs old and can set its own clock to it as the clock slave.

- The high-precision time synchronization of all communication nodes is a requirement for division of the bandwidth and the transmission times in the red and green phase.
- In each cycle, the synchronous data is first transmitted in the **red phase**. Protected from delays caused by other data and allows a high level of determinism.
- The “open” **green phase**, all other data are transmitted according to IEEE 802.1Q and the specified priorities.

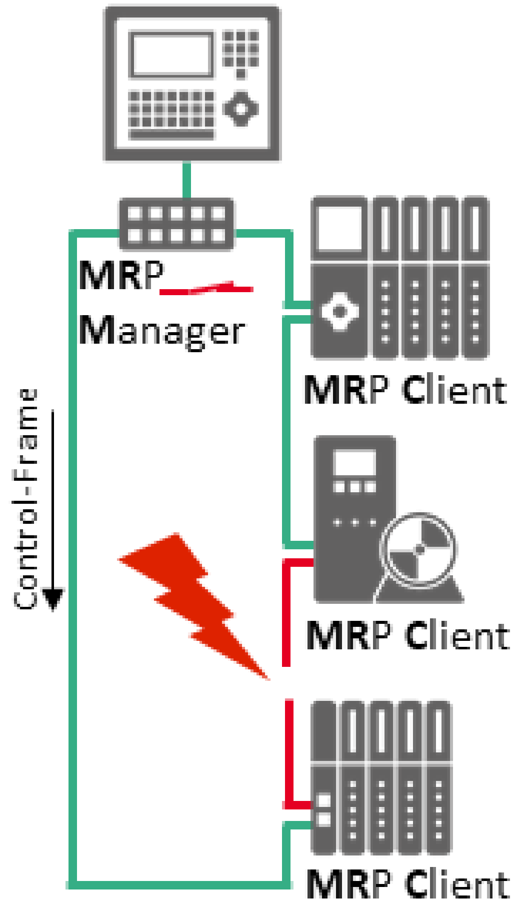
Higher Availability through Media Redundancy

- Media Redundancy Protocol (MRP)
  - IEC 62439-2
  - Reconfiguration time of less than 200 ms for communication paths with TCP/IP and RT frames after a fault.
  - Error-free operation of an automation system involves a Media Redundancy Manager (MRM) and several Media Redundancy Clients (MRC) arranged in a ring.
    - The MRM checks the functional capability of the configured ring structure by sending out cyclic test frames. As long as it receives all of its test frames back, the ring structure is intact.
    - In case of a fault, the redundancy manager logically opens the ring. Through this behavior, an MRM converts a ring structure into a line structure and thus prevents the circulation of frames.

Higher Availability through Media Redundancy



In normal operation the ring topology is reduced to a line



In case of failure the ring topology is reduced to a line again

Higher Availability through Media Redundancy

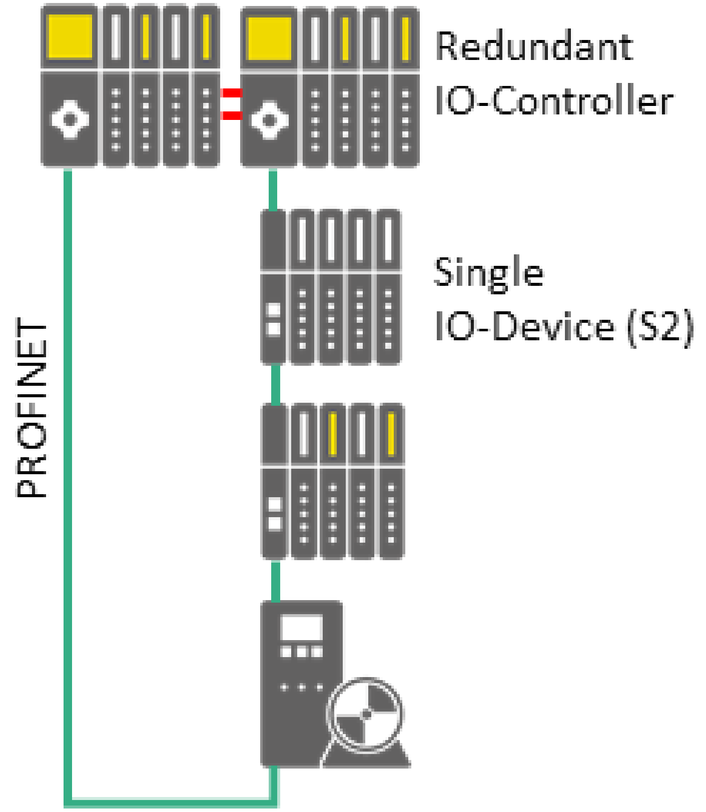
- Media Redundancy for Planned Duplication (MRPD)
  - IEC 61158 for IRT communication
  - Enables smooth switchover from one communication path to another in the event of a fault and is used together with MRP.
  - This concept is based on ring redundancy, where the IO controller sends the IO data to the devices in both directions of the ring at the same time.
  - During system power-up, the IO controller loads the information of the communication paths for both communication channels (directions) into the individual nodes in the ring.
  - Loading of the "schedule" alone is sufficient to exclude frames from circulating in this variant: the recipient rejects the second (duplicate) frame.

Higher Availability through System Redundancy

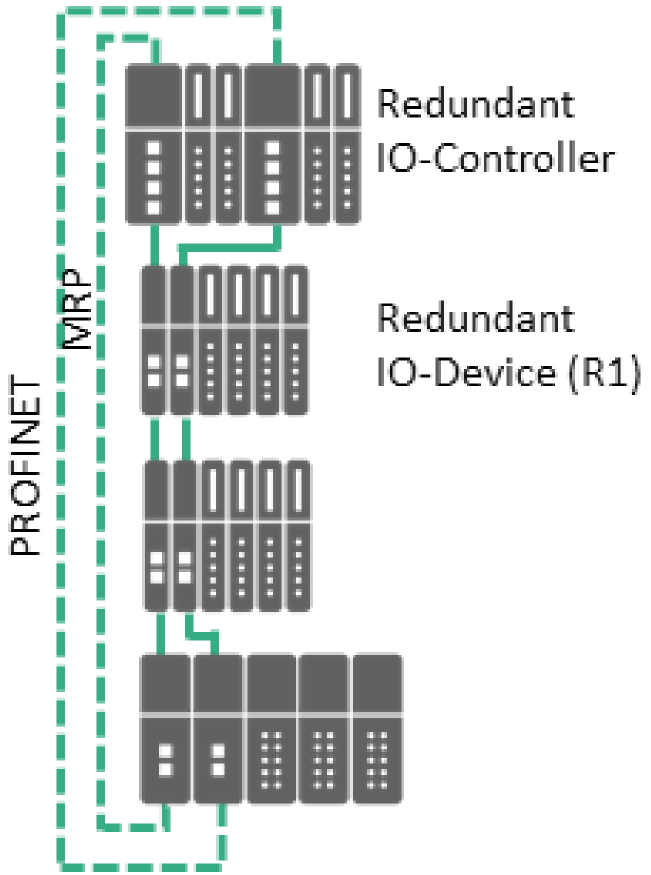
- Doubling of critical components – the controllers
- If a controller fails, the second controller takes over operation within a very short period of time, usually without interrupting the process.
- For redundant communication between the controllers and the devices, PROFINET offers a scalable solution.

S2 redundancy

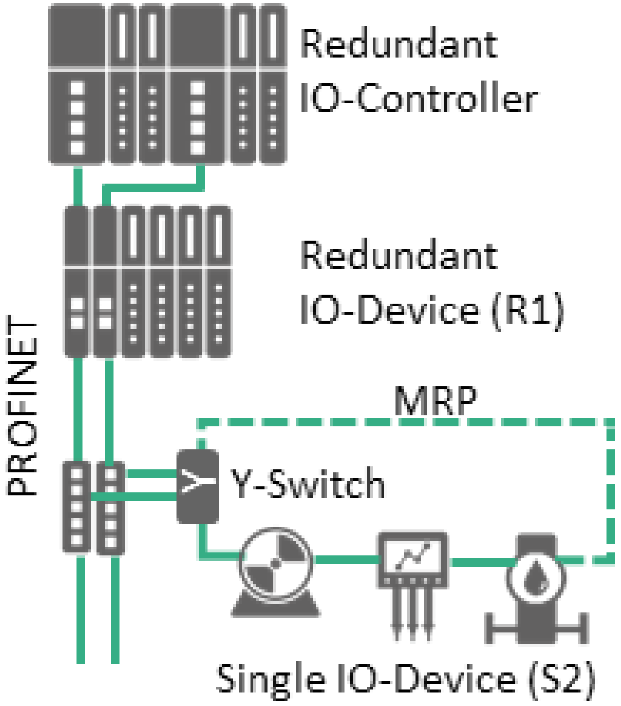
- Simple and easy communication with a redundant controller is achieved with S2 redundancy.
- The redundant IO controller is connected at the respective end of a PROFINET line. The IO devices establish two connections to the redundant controller: one to the "left" and one to the "right."
- One of the connections is active and connects the IO device to the active part of the redundant controller. If the active connection is interrupted or the active IO controller fails, switching to the second connection occurs in a very short period of time.



System Redundancy (S2)



System Redundancy (R1)



System Redundancy (R1, S2, MRP)

R1 redundancy

- Two PROFINET networks are installed. One network is assigned to each part of the redundant IO controller.
- The IO devices feature a separate communication interface for each network, i.e. their communication interfaces are designed for redundancy.
- To increase availability even further, both PROFINET networks can also be implemented as MRP rings. This enables the system to handle more than one fault, to an extent.

isep Instituto Superior de Engenharia do Porto

P.PORTO

INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Rua Dr. António Bernardino de Almeida, 431

4249-015 Porto, Portugal

—

T(+351) 228 340 500

F (+351) 228 321 159

—

www.isep.ipp.pt

www: moodle.isep.ipp.pt

http://www.isep.ipp.pt/mescc

e-mail: llf@isep.ipp.pt

AENOR ER Empresa Certificada ISO 9001

IQNet CERTIFIED MANAGEMENT SYSTEM

cdio CONCEIVE DESIGN IMPLEMENT OPERATE

Study in Europe

EUR-ACE®