EtherCAT Relay function

Karl WEBER
Marcel KIESSLING

Aug 2022

Disclamer

This presentation should be considered as the personal views of the presenters and not as a formal position, explanation, or interpretation of IEEE and ETG.

More details can be found in the presentation from 2017:

https://www.ieee802.org/1/files/public/docs2017/liaison-ETG-streamAdaption-1117.pdf

The website of the EtherCAT Technology Group (ETG) provides further information about EtherCAT: https://ethercat.org/en/tech_group.html

Per IEEE-SA Standards Board Bylaws, August 2020: At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

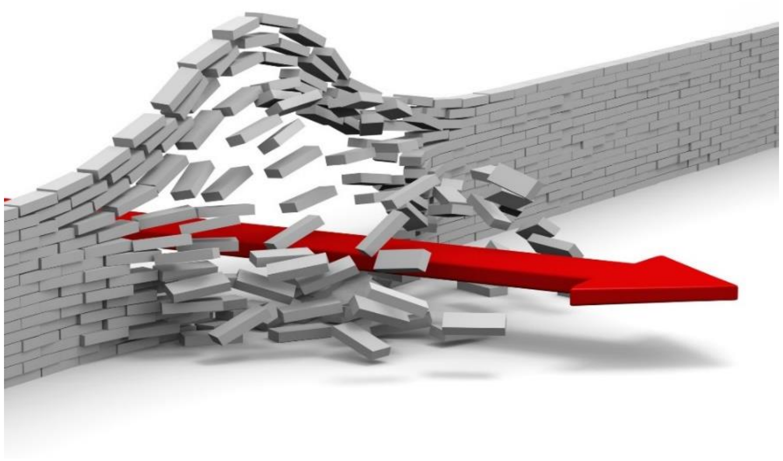
IEEE 802 challenge for high speed applications

- Efficiency: low byte count (8 bytes) needed vs. 84 octets minimum for Ethernet
- Delay: cable delay of fieldbusses vs. store and forward/bridging
  (passive media used for first solutions)
  + interfering traffic

➔ **Overall efficiency 3%**
(with 128 octets interfering traffic)

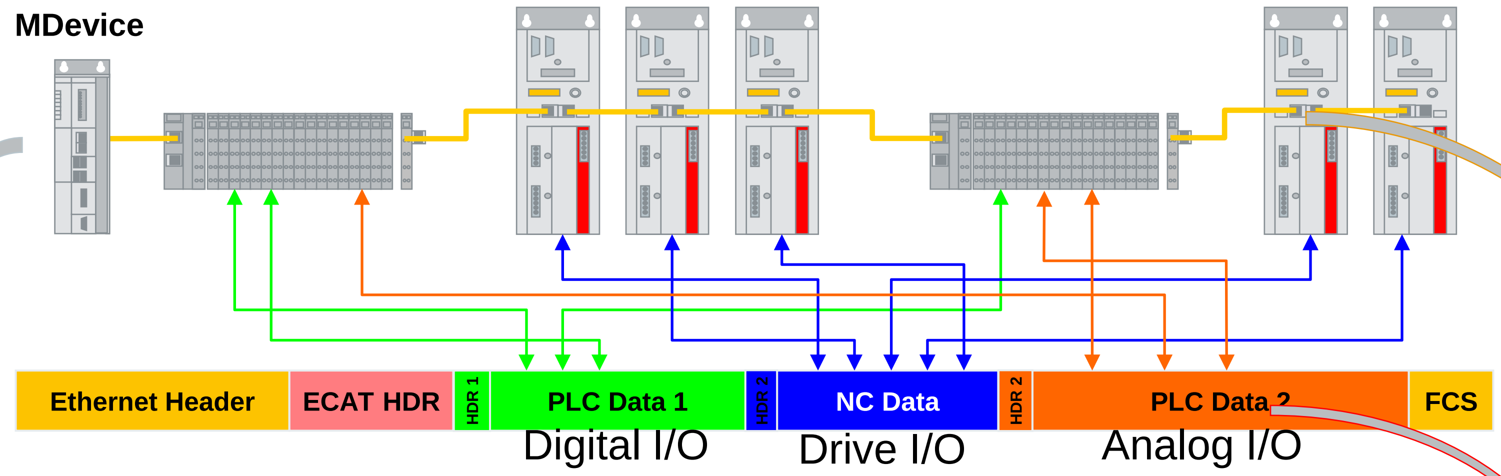
- This leads to the **EtherCAT** approach

EtherCAT®



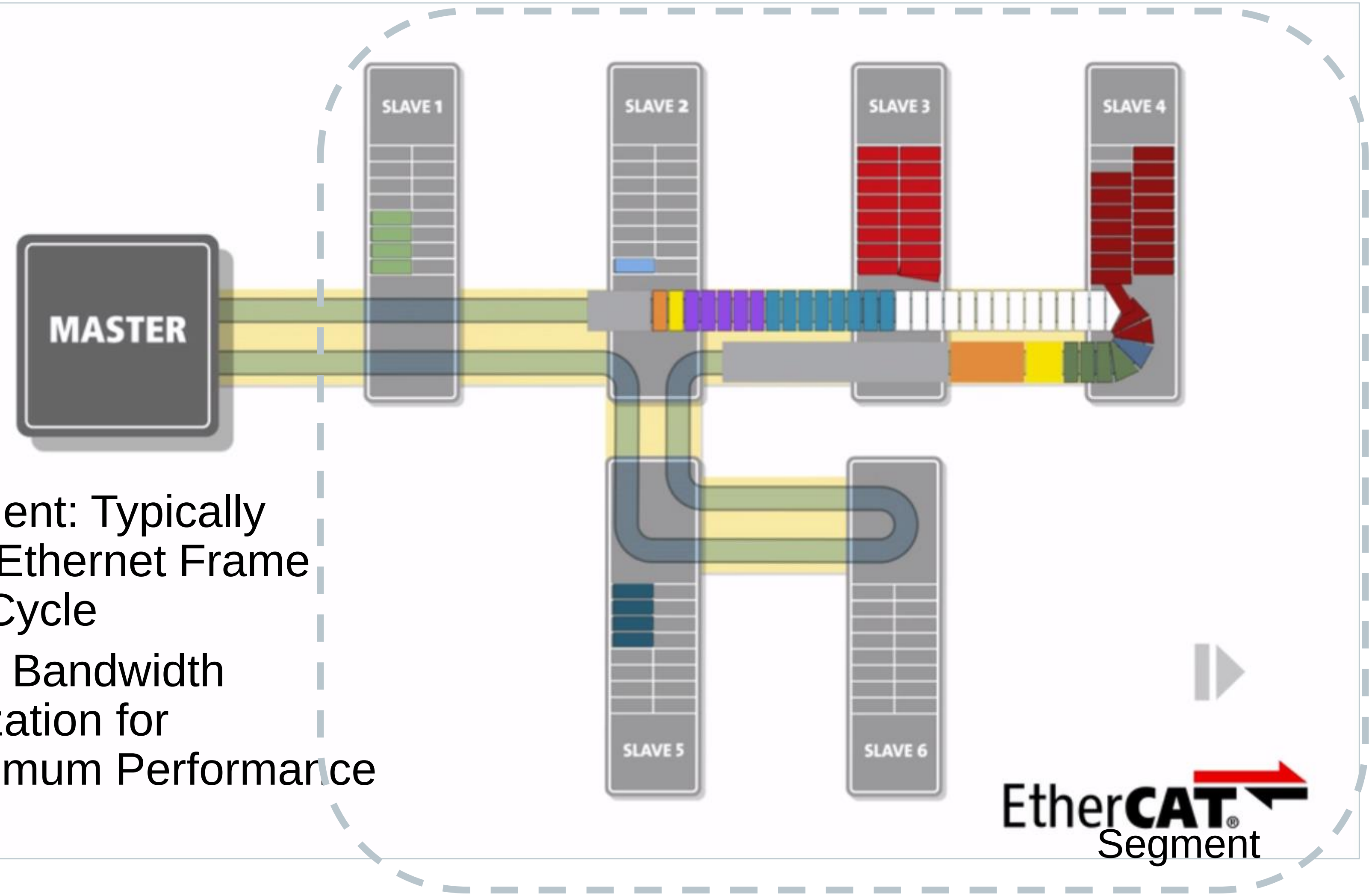
- Efficiency **Shared frame** instead of individual frame
  ➔ performance improvement: overhead 50 Bytes instead of 750/1500
  … in a network of 10/20 I/O stations
- **Processing on the fly** with topological relay function (automatic)
  Instead of address based forwarding
  ➔ performance improvement: 0,6µs instead of >3µs (7µs/store&forward)

Functional Principle | Ethernet „on the fly“



- Ethernet-compatibility maintained
  - MDevice Implementation on standard Ethernet interface (MAC-Interface)
  - Standard PC or Embedded PC sufficient - no dedicated plug-in card on-board Ethernet Port is fine
- Minimal overhead (= shared frame)
  - Optimized frame structure for I/O modules connected
  - L2 Communication in hardware within segment: **predictable** performance
  - No bridging, just relaying to next station in the loop …….

Functional Principle: Ethernet “on the fly”



- Efficient: Typically one Ethernet Frame per Cycle
- High Bandwidth Utilization for maximum Performance

*Master/Slave shall be replaced by MDevice/SubDevice

Animation available as EtherCAT Functional Principle (2D) on https://www.youtube.com/watch?v=z2OagcHG-UU

Preconditions for the segment forwarding

- Same Data Rate within the segment (typical 100 Mb/s – 1000 Mb/s possible)
- The MDevice will send out frames and receive frames
- Each SubDevice in the Segment just relays frames to the next port
- The forwarding acts as a unidirectional relay from port to port
    - Receive Port 0 connected to a virtual Port to allow a DLL entity to put information in the frame and get information „on the fly“
    - The forwarding takes place on any open port
    - The forwarding already starts with the Preamble
- If there is only one port open the relay function will send it back at the same port
- Each device that is connected will receive the frame
- The flow of data in a segment forms a logical ring of all devices connected

➔ The ports of a node are ordered as a ring with Port0 as first entry

➔ If the main structure is a line, the exits of the line shall be in between Port0 and Port(last) for easier handling and diagnostic (no technical reason)

Data Flow - Example

Backbone 1

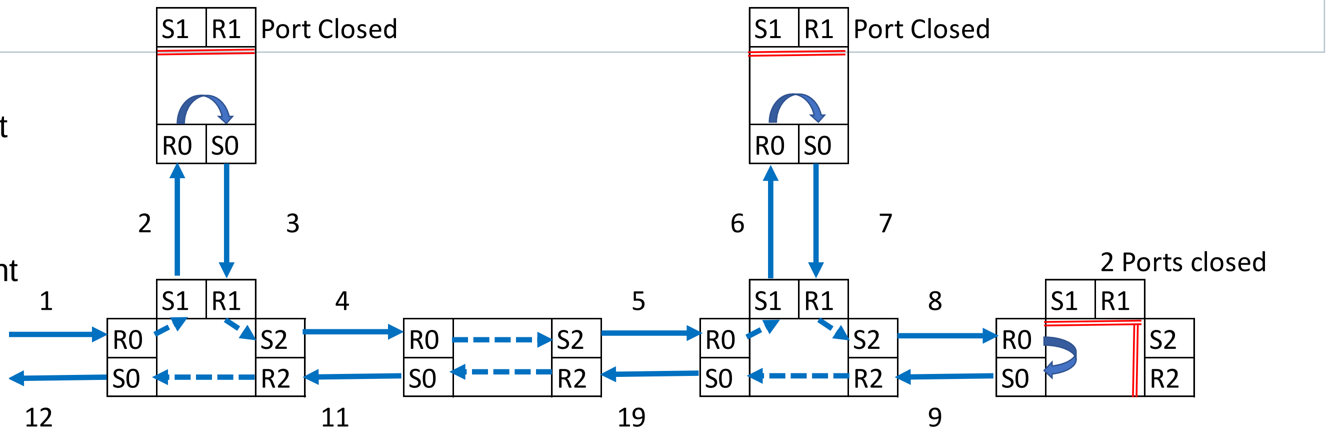
1st Sub-segment

Backbone 2

Backbone 3

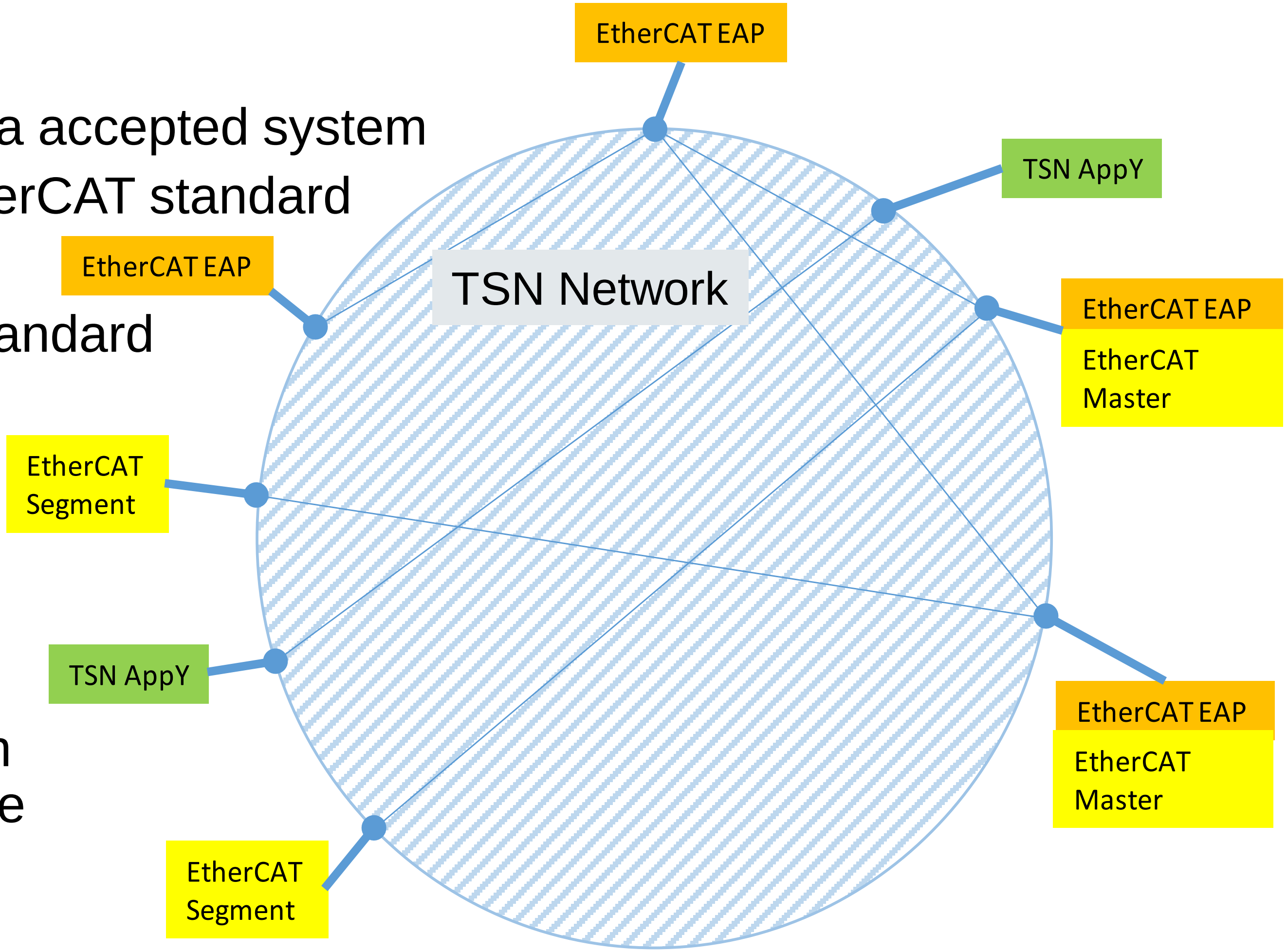
2nd Sub-segment

Backbone 4

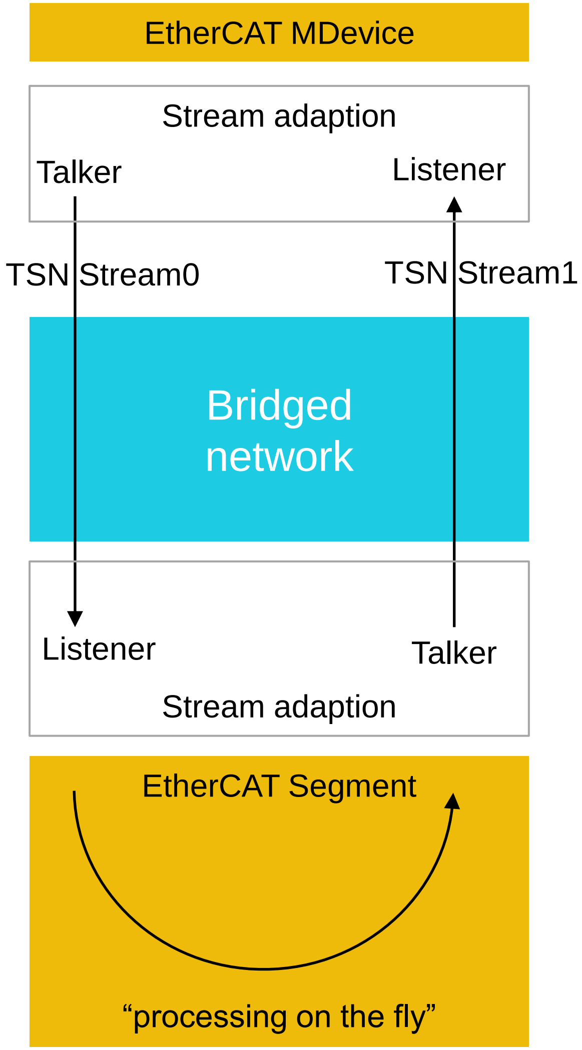


Use of Open mode → TSN

- Idea
  - Never touch a accepted system
  - Maintain EtherCAT standard
  - Use 802.1 standard
- Compatibility with other 802.1 usage
- Specifies adaptation

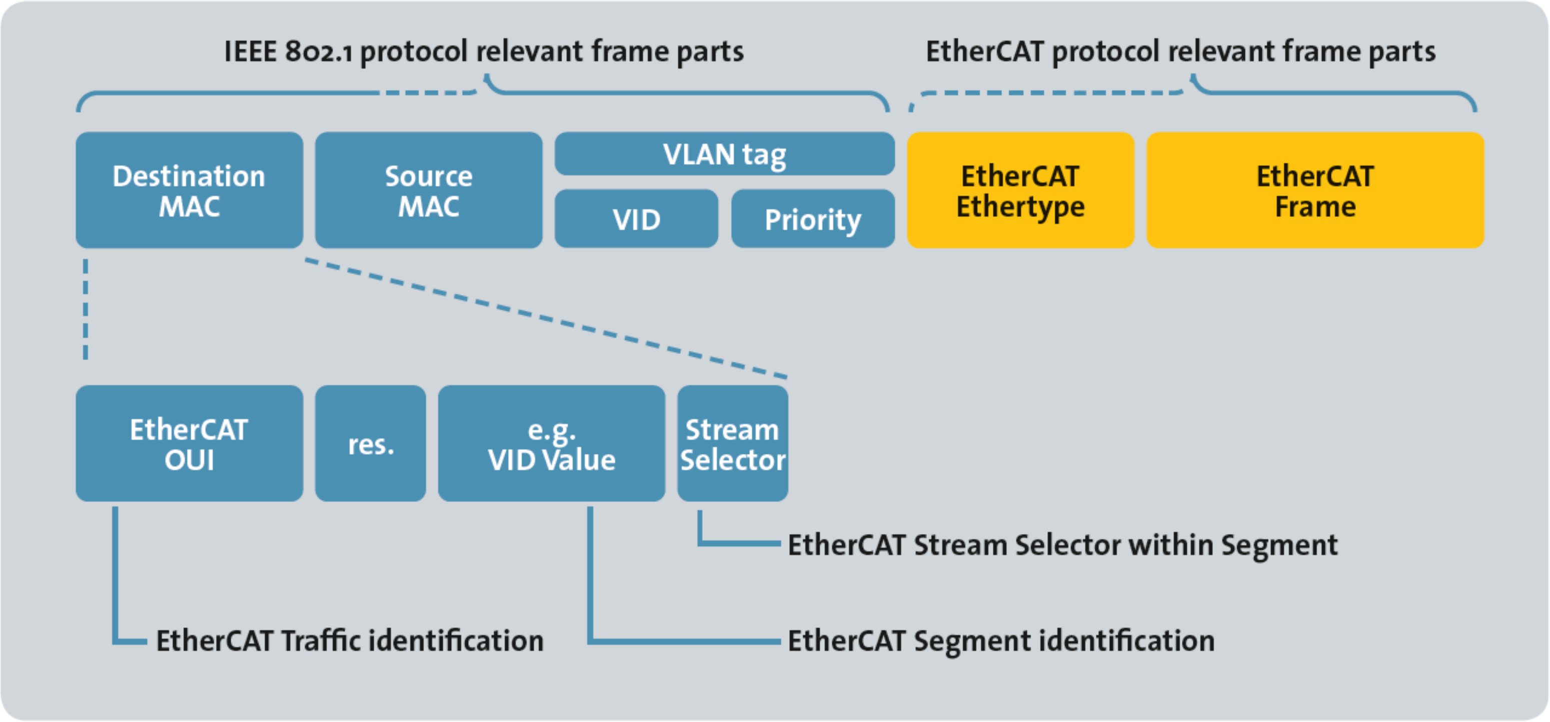


Stream Adaptation: Details



- Always a pair of streams is set up
- Minimum one pair, but more might be set up, e.g.
  - One for cyclic
  - One for acyclic (strict priority)
  - for additional transfers
- Traffic class for pair of stream always the same
- Maintain Traffic Class (VLAN Prio)
- Maintain length
  (EtherCAT Rx/TX frame length identical)
- **Unique Stream Identification required(!)**

Protocols use different fields


IEEE 802.1 protocol relevant frame parts
EtherCAT protocol relevant frame parts
Destination MAC
Source MAC
VLAN tag
VID
Priority
EtherCAT Ethertype
EtherCAT Frame
EtherCAT OUI
res.
e.g. VID Value
Stream Selector
EtherCAT Stream Selector within Segment
EtherCAT Traffic identification
EtherCAT Segment identification

Open Mode

- EtherCAT segment corresponds to an Identifier (“VID”)
- MAC addresses for stream identification (StreamDA, StreamSA) constructed of
  - OUI, (V)ID, Stream selector
- Multicast DAs are possible as stream MAC for TSN Networks
  Null Stream Identification combined with Source MAC and VLAN Stream Identification according to Table 6-1 of 802.1CB
- Unicast DAs are possible for streams to the segment
  Address change for backward direction to avoid multicast scans in the MDevice and enables address learning in network