NetTimeLogic GMBH

Introduction to Time-Sensitive-Networking (TSN)

TSN Basics

Content

- What is TSN?
- Why do we need/want TSN?
- How does TSN work?
- TSN and OPC-UA
- Migration paths towards TSN/OPC-UA
- Why are FPGAs/SoCs the best choice for TSN?
- An implementation of TSN on an FPGA/SoC
- Summary

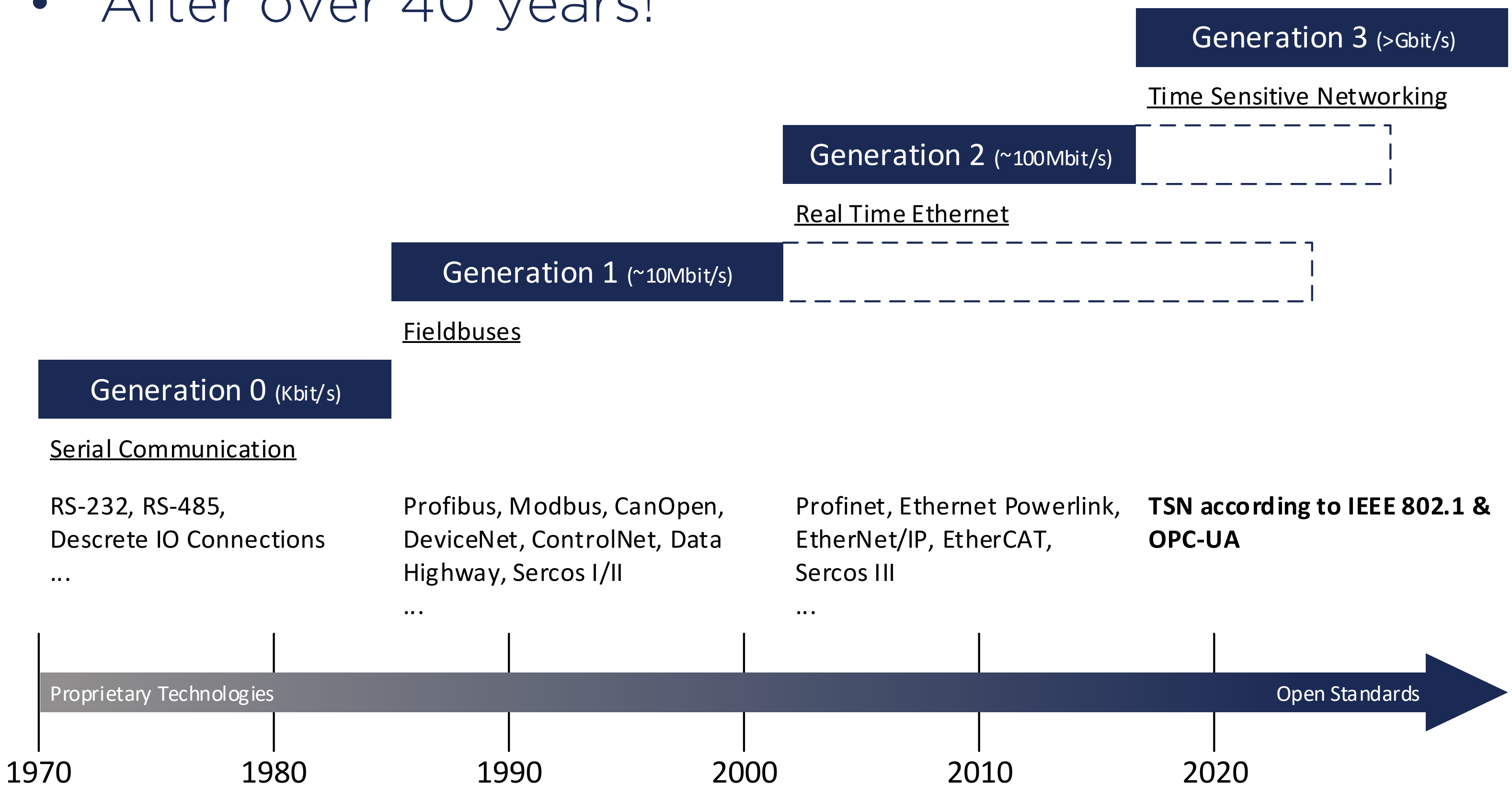
What is TSN?
Overview

- **Extensions to normal Ethernet**
  - Synchronization
  - Prioritization
  - Determinism and Bounded Latency
  - High Availability
  - Process Network and Office Network combined

- **Basically what the Real Time Ethernet Solutions (e.g. Profinet, EtherCAT, …) do today, but in a open standardized manner**
  - Set of IEEE 802.1 Standards

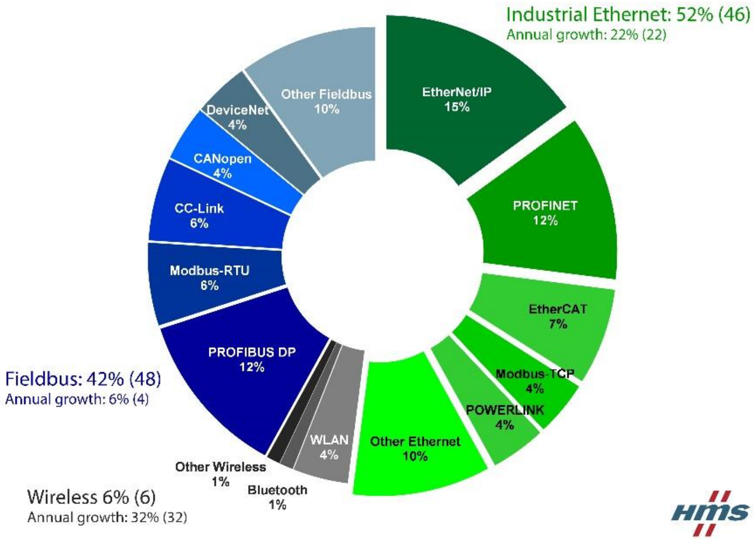
Why do we need/want TSN?
History of OT Communication

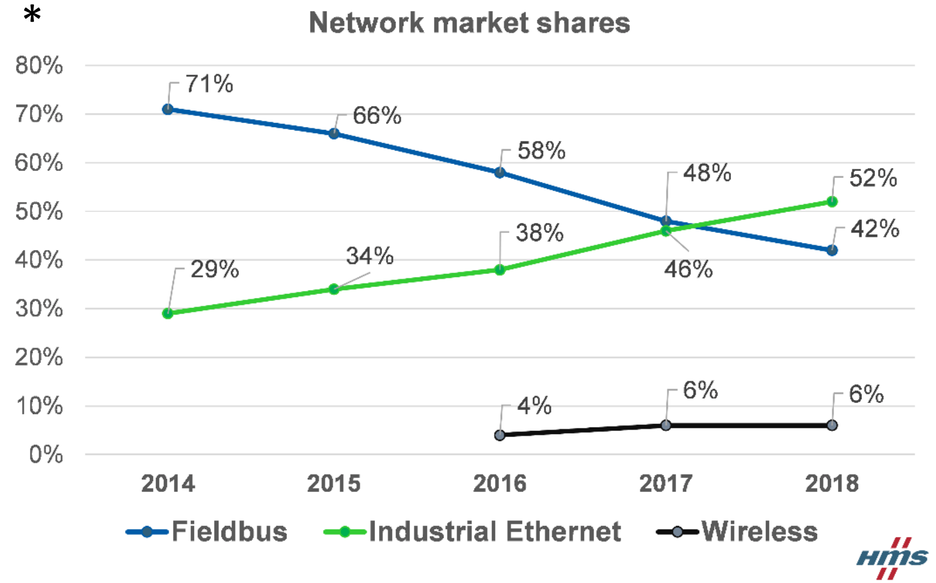
- **We want to end the war on field communication!**
  - After over 40 years!



Why do we need/want TSN?
History of OT Communication

- **Industrial Ethernet took over traditional Fieldbuses in 2018 and is growing!**
  - The trend to Ethernet (as base for TSN) is there

*


*


Why do we need/want TSN?

Drivers for TSN

- **Open Standards for Communication**
  - Not driven by one large company
    - Unlike e.g. Profinet (Siemens), EtherCAT (Beckhoff)…
  - Vendor independence
- **Easy migration path from existing networks**
- **Interoperability is a key driver for TSN**
- **ONE Network for everything**
  - Best effort and critical data over one network
  - Vertical integration from the cloud to the processing

Why do we need/want TSN?
TSN for automation

NetTimeLogic GMBH

- **Will be THE Ethernet based field bus of the future**
  - Enabler for Industry 4.0 and IIOT
  - Vertical Integration with TSN from the cloud to the factory floor possible

- **Sensor-, actor- and control-equipment vendors don't need to create a version of their products for each field bus**
  - One fits all
  - Cheaper

Why do we need/want TSN?

TSN for automation

- **Customers don't need to choose a specific field bus**
  - Better interoperability
  - Multi source
    - Competition between equipment vendors
    - Cheaper
  - No gateways needed
  - Reduced complexity

Why do we need/want TSN?

TSN for automotive

- **Will be the onboard bus of the future**
  - Enabler for autonomous cars
- **Automotive Ethernet will replace the old CAN bus**
  - High bandwidth, low latency, high availability and determinism are key requirements which are all addressed by TSN
  - Combined Advanced Driver Assistance Systems (ADAS) with Infotainment and "normal" Car control over one Network thanks to TSN
  - Cleaner and cheaper cabling

Why do we need/want TSN?

TSN for utilities

- **Some of the key features are already there**
  - E.g. Synchronization, Network redundancy
- **Especially bounded latency and the possibility to mix a "process bus" with a "station bus" in a standardized manner by still fulfilling all requirements are drivers for the utilities industry to converge to TSN in the future.**

How does TSN work?
Overview

- **There is NOT ONE TSN standard**
  - A set of IEEE 802.1 standards which together form TSN
- **OSI Layer 2 functionalities**
- **Common-sense what the minimum set of functionalities is to call it TSN is:**
  - Ethernet (Transport medium)
  - Time Synchronization
  - Traffic Classification (Priority Handling)
  - Scheduled Traffic (Cycles and Time Slots)

How does TSN work?
Priority Handling

- **IEEE 802.1Q**
  - VLAN (QoS)
- **Distinguish between high priority, reserved and best effort traffic classes (and even more)**
  - Know which traffic is important and has hard real-time requirements, which traffic might have soft real-time requirements and which traffic is just best effort traffic
- **It is the base for everything else in TSN**
  - Most of the other standards are Amendments to IEEE802.1Q (e.g. Qbv, Qch, Qav etc.)

How does TSN work?
Synchronization

- **IEEE 802.1AS (or also IEEE 1588)**
  - Time Synchronization
- **Common time in every TSN node**
  - Which is the base for all other mechanisms in TSN
- **Precision Time Protocol allows phase and frequency synchronization over Ethernet**
  - Sub microsecond accuracy
  - Widely used already (IEEE 1588)
- **Every node (e.g. also Switches) must support PTP**
  - Boundary Clocks (multiport, switches, routers)
  - Ordinary Clock (end nodes)

How does TSN work?

Scheduling

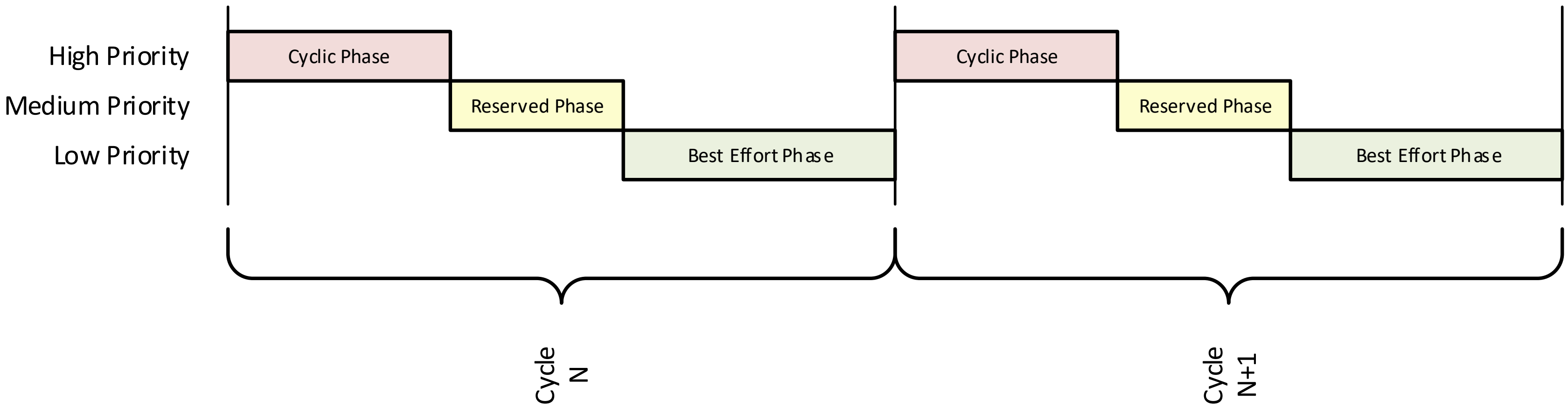
- **IEEE 802.1Qbv**
  - Cycles and Time Slots
- **Based on the synchronized time, the transmission is divided into cycles and time slots**
- **Defines when a frame is allowed to be sent depending on the traffic class**
  - Reserved time slots
  - Guaranteed time slot to send a frame
    - Best effort traffic will not block the network for real-time traffic
- **Can change order of frames (between traffic classes)**

How does TSN work?
Scheduling

- **Hold a list with entries which traffic class is allowed to send and for how long**
  - Repeated every cycle (cycle start can also be defined)
  - Multiple traffic classes can be active at the same time
  - Frames shall not exceed the end of a phase of a specific traffic class
    - Check if a frame can be sent before phase end
    - A phase can span over multiple entries (this has to be taken account for in the end of phase calculation)

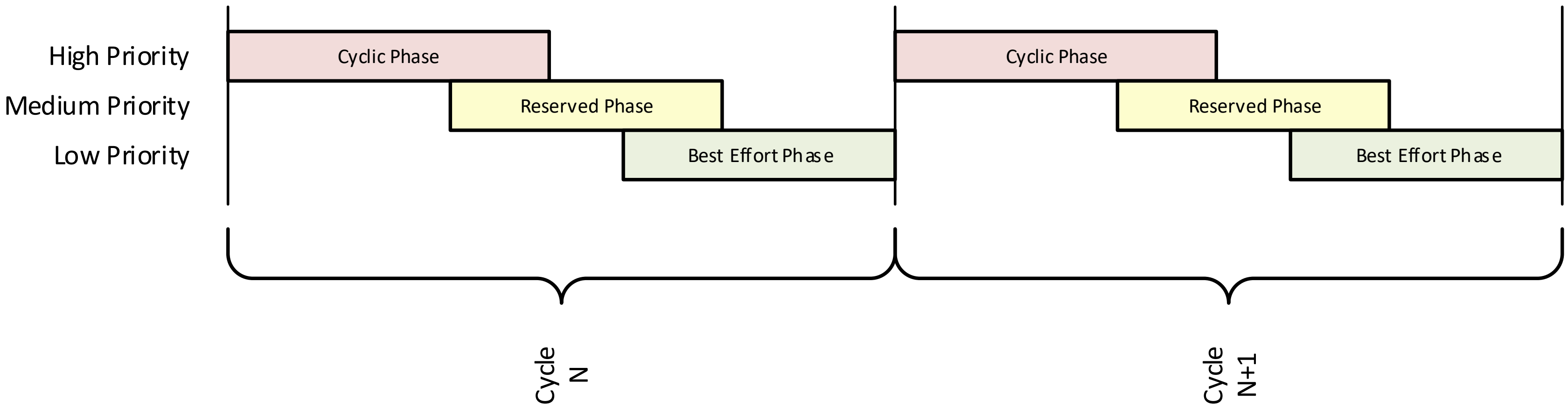
How does TSN work?
Scheduling

- **Exclusive access**
  - Only one traffic class per time
  - Frames must end before the end of a phase
    - Guard Bands might be required



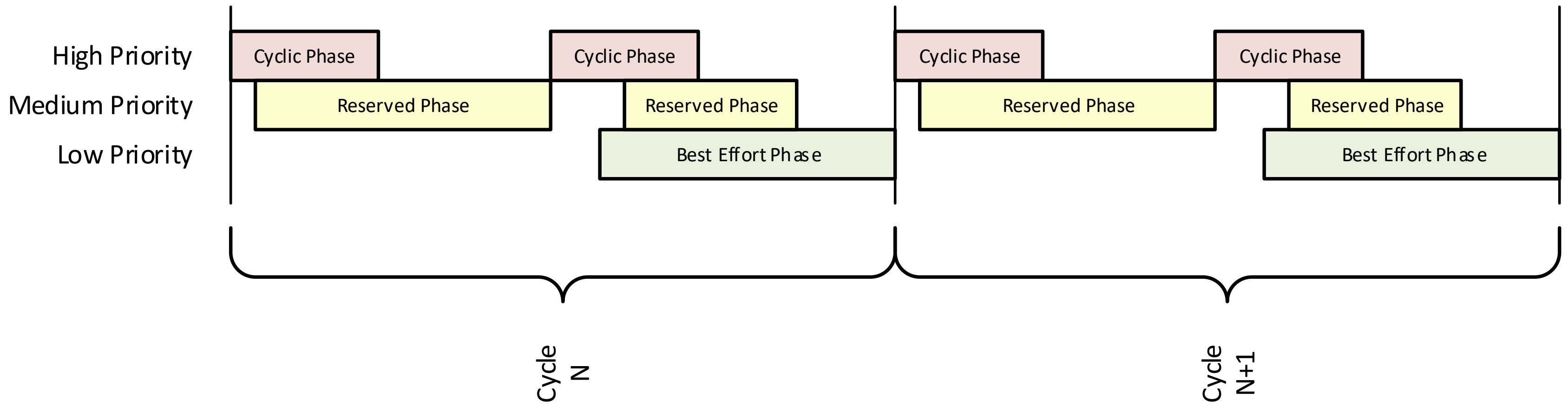
How does TSN work?
Scheduling

- **Overlapping access**
  - Frames must end before the end of a phase
  - Priority scheduling during overlap



How does TSN work?
Scheduling

- **Multicycle access**
  - Frames must end before the end of a cycle
  - Priority scheduling during overlap
  - Multiple phases of same class per cycle



How does TSN work?
Preemption

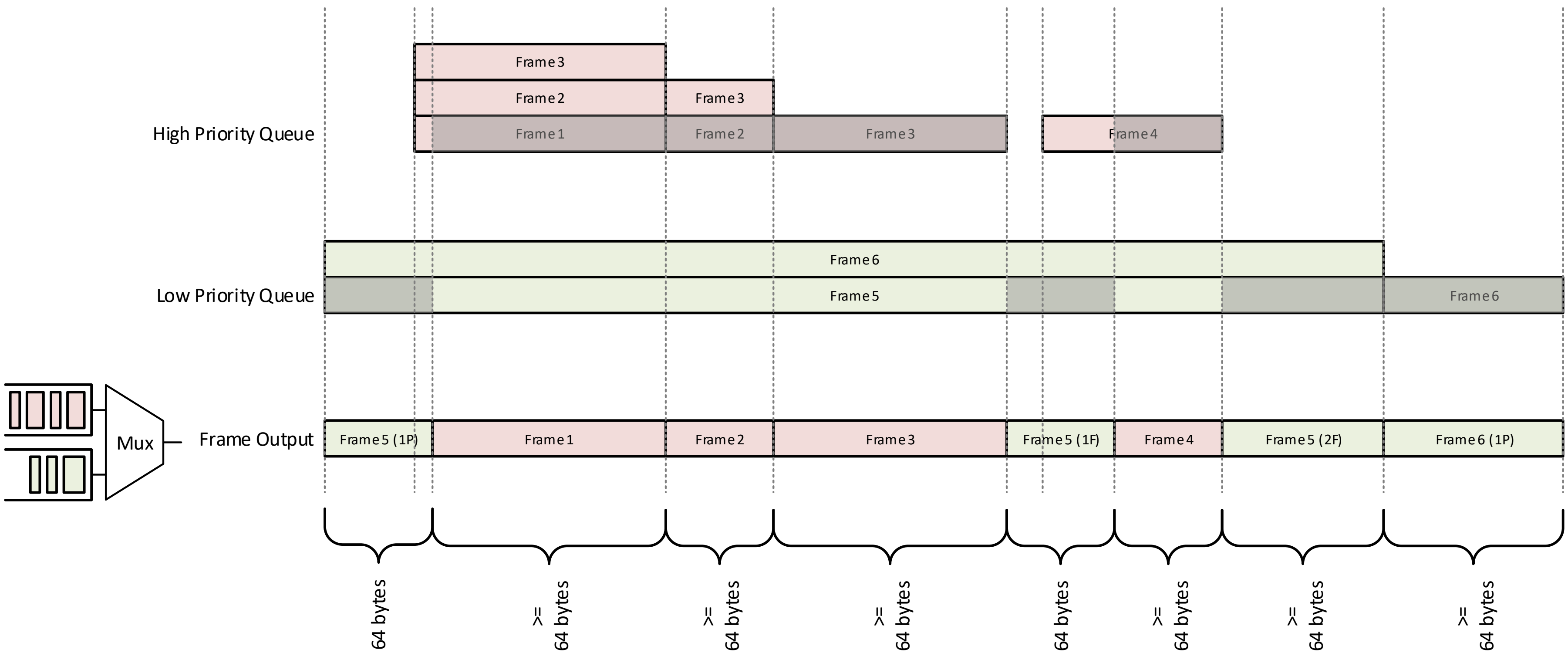
- **IEEE 802.3br & IEEE 802.1Qbu**
  - Interrupting/preempting frames
- **High priority frames can interrupt low priority frames**
  - Low priority frame is stopped the high priority frame sent and the low priority frame resumed
  - A low priority frame can be split into multiple fragments
  - Only one frame can be preempted at the time
- **Used when multiple traffic classes can be active at the same time or no scheduling is done**

How does TSN work?
Preemption

- **Special Preamble an Start of Frame Delimiter (SFD) and modified CRC to detect preemptable and preempted frames**
  - Frame counter as part of Preamble & SFD
  - Modified CRC per fragment (added)
- **Minimal Ethernet frame size is preserved**
  - Min 64byte fragments (60bytes plus MCRC), max wait delay of high priority frames
  - Frames <124 can not be preempted
- **Splitting done at sender, assembly of preempted frames at receiver**
- **Will change the order of frames**

How does TSN work?
Preemption

- Example, two traffic classes, simultaneous sending



How does TSN work?
Cyclic Forwarding

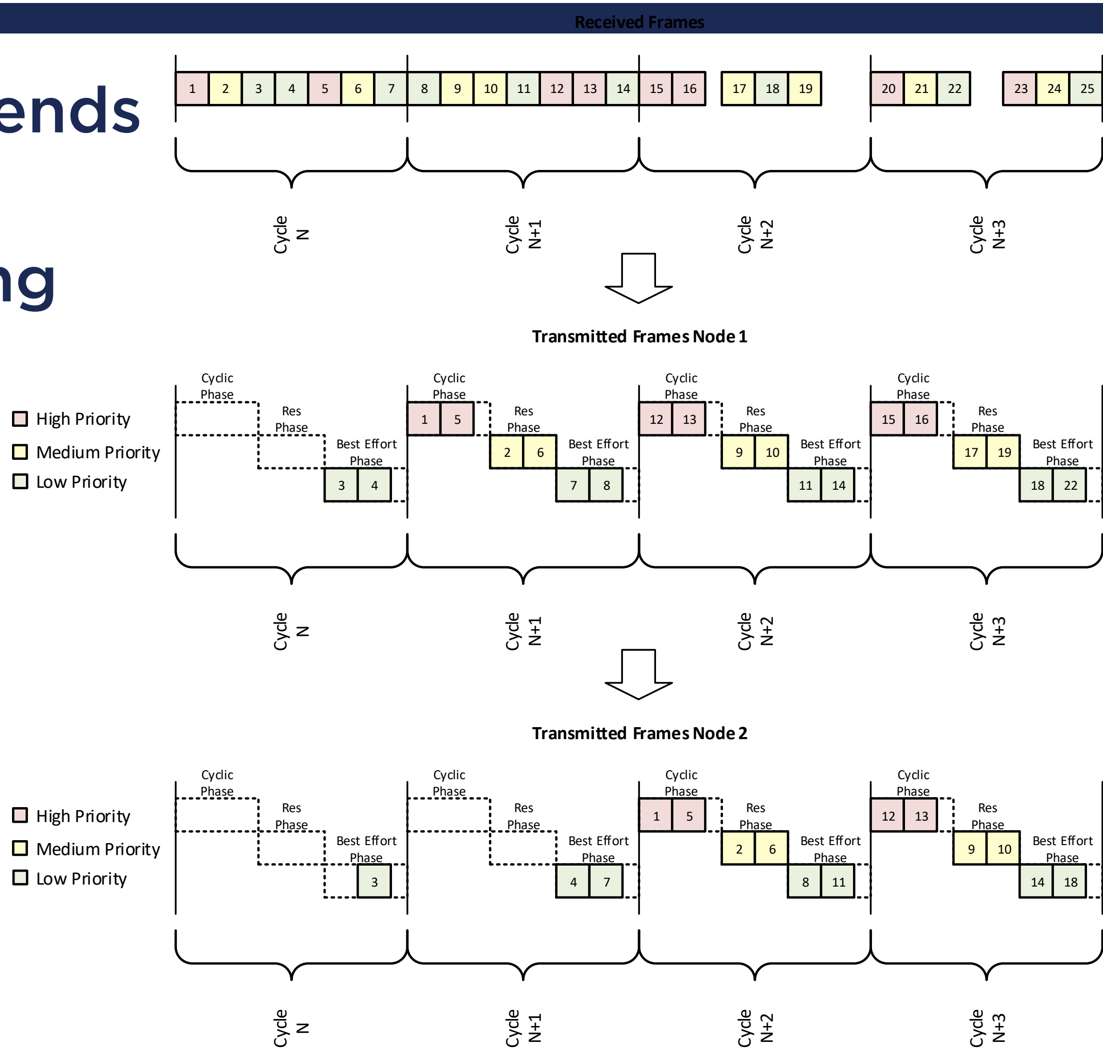
NetTimeLogic GMBH

- **IEEE 802.1Qch**
  - Deterministic forwarding
- **The delay depends on number of hops**
  - Each hop stores the frames received in one cycle and forwards it in the next cycle
- **Maximum delay of each communication path can be easily calculated for real-time frames**
- **Cyclic Forwarding normally not done for best effort traffic**
- **Frames will come as bursts in the next cycle**
  - Can change order of frames between different traffic classes

How does TSN work?

Cyclic Forwarding

- Non TSN node sends mixed traffic
- Cyclic Forwarding combined with Scheduling
- Best effort forwarded when possible (no Cyclic For-warding)



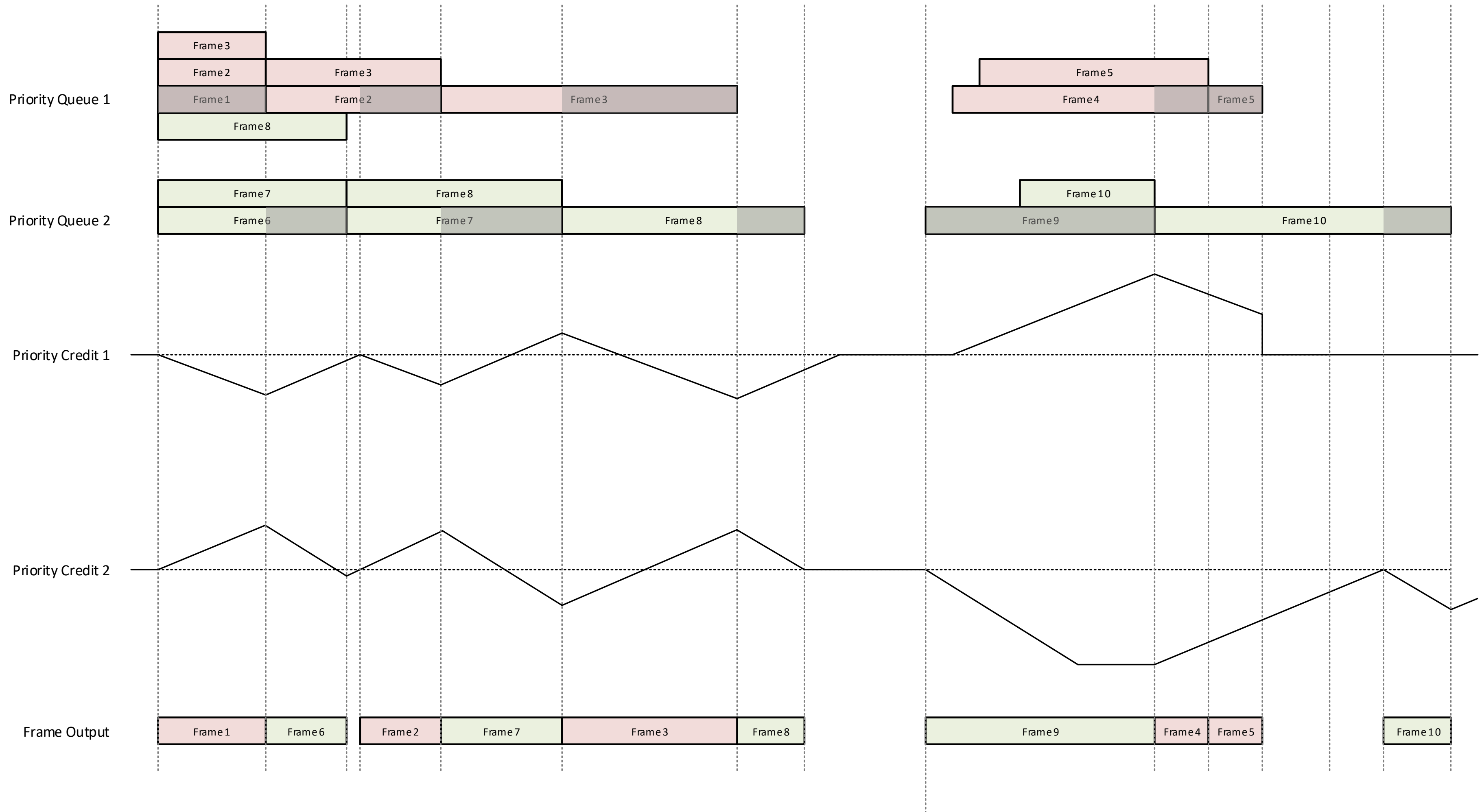
How does TSN work?
Traffic Shaping

- **IEEE 802.1Qav**
  - Credit Based Shaper
- **Two functionalities in one**
  - Avoid bursts of frames of the same traffic class or stream
  - Change priorities between different traffic classes or streams
- **Per stream or traffic class**
- **Can change order of frames**
  - Between different traffic classes
  - Between different streams

How does TSN work?
Traffic Shaping

- **Algorithm**
  - Decrement credit during sending (depending on frame length)
  - Increment credit when sending has to wait
  - Allow sending only when credit is >=0
  - Max and Min credit values for maximum wait time and maximum burst lengths
  - When credit >0 and not waiting to send set credit to 0
  - Increment and Decrement can be chosen

How does TSN work?
Traffic Shaping



How does TSN work?
Filtering and Policing

- **IEEE 802.1Qci**
  - Filtering and Policing of incoming frames
- **Protection against excess bandwidth usage, burst sizes as well as against faulty or malicious endpoints**
- **Fault isolation to one segment and not the whole network**
- **Filter frames on ingress port based on:**
  - Arrival times
  - Rates
  - Bandwidth

How does TSN work?
Network Redundancy

- **IEEE 802.1CB**
  - Network Redundancy
- **Seamless redundancy (zero loss)**
  - Duplicating frames on sending to multiple paths
  - Duplicate rejection on reception (take first)
  - Redundancy Tagging of Frames
- **Can be HSR or PRP according to IEC 62439-3 or the scheme defined in IEEE 802.1CB which is basically the same as HSR**

How does TSN work?
Path Control and Reservation

- **IEEE 802.1Qcc (& IEEE 802.1Qca)**
  - Stream definition, Scheduling and Bandwidth configuration
- **A scheme to configure streams in TSN nodes and their characteristics:**
  - Time Slots
  - Bandwidth
  - Max frame sizes
  - Priorities
- **Talkers and Listeners say what they want**
  - Centralized or Decentralized configuration

How does TSN work?
Configuration and Supervision

- **NETCONF/RESTCONF&YANG vs SNMP&MIB vs OPC-UA vs LLDP**
  - Configuration and supervision of the TSN nodes
- **Multiple schemes are currently present in the TSN implementations**
  - Not 100% clear which scheme will win
    - Probably NETCONF/RESTCONF&YANG for the Network Infrastructure
    - Probably OPC-UA for End Nodes (since it is there?)
  - Big topic in the testbeds and standardization

How does TSN work?

And now?

- **All of those TSN standards combine to a very powerful toolset**
  - All standards can be combined as required
  - It really depends on the application which standards you require
- **Which standards are mandatory?**
  - Profiles are in standardization phase to define which are mandatory and optional for different industries (e.g. automation (IEEE/IEC60802), automotive, …)

TSN and OPC-UA

TSN and Application Layers

NetTimeLogic GMBH

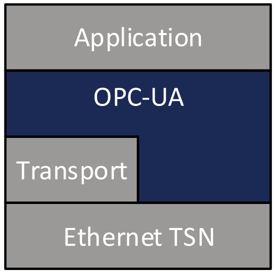
- **TSN handles only OSI Layer 2**
  - Unlike Profinet, EtherCAT etc. which define also the communication and transport level with object modeling and machine-machine communication
- **Existing Real Time Protocols have migration paths to use TSN as lower layer together with the existing "proprietary" higher layer**
  - Profinet over TSN
  - Sercos over TSN
  - EtherCAT with TSN
  - EtherNet/IP over TSN
  - etc.

TSN and OPC-UA
Overview

NetTimeLogic GMBH

- **There is an Open Standard alternative**
  - Open Platform Communications Unified Architecture (OPC-UA)
  - Will replace in the future the “proprietary” higher layers of the existing Real Time Ethernet protocols
- **Close collaboration between the TSN working group and the OPC foundation**
  - Together the build a complete field bus solution, with TSN as the lower level and OPC-UA as the higher level

TSN and OPC-UA
Overview



- **OPC-UA defines two main things**
  - Transport mechanisms
  - Data modeling
- **Two main communication schemes**
  - Client/Server 1:1 (Configuration in TSN)
  - Publisher/Subscriber 1:N (Data Streams in TSN)
- **Work in progress**
  - Most functionalities are already there
  - Extensions coming bit by bit
- **Many open source implementations of OPC-UA**
  - In ANSI C, C++, Java, .NET, etc.

Migration path towards TSN
Overview

NetTimeLogic GMBH

- **Commitments of the big companies behind the existing Real Time Ethernet solutions to migrate to TSN and OPC-UA (@TSNA conference)**
  1. Gateways between existing Real Time Ethernet solutions and OPC-UA over TSN (partly done)
  2. Replace the lower layer of the existing Real Time Ethernet solutions with TSN (partly done)
  3. Replace the higher layers of the existing Real Time ethernet solutions with OPC-UA (in progress)

- **The way is open for TSN/OPC-UA field installations!**

Migration path towards TSN
Interoperability Testing

NetTimeLogic GMBH

- **Priority number one is interoperability!**
  - The whole ecosystem will only be successful when interoperability is guaranteed

- **Several TSN (and OPC/UA) Testbeds to test interoperability**
  - IIC TSN Testbed (link)
  - LNI 4.0 TSN Testbed (link)
  - Huawei TSN Testbed (link)
  - UNH-IOL TSN Test Infrastructure (link)
  - Etc.

Why are FPGAs/SoCs the best choice for TSN?

Overview

- **Some of the TSN standards are still in draft state or work in progress and there are more standards/functionalities to come**
  - ASICs only for a small set of finished standards (e.g. 802.1AS, 802.1Qbv)
  - FPGA Solutions can easily adapt to standard changes and can be extended with upcoming standards/functionalities

Why are FPGAs/SoCs the best choice for TSN?

Overview

- **Real Time Data talkers and listener with very short cycle times (e.g. 15.625us, 31.25us, 62.5us) can be challenging in Software**
  - Real Time Data Talkers and Listeners in software will generate high CPU load and/or require powerful CPUs (expensive)
  - Real Time Data Talkers and Listeners can be implemented directly in the FPGA, allowing minimal cycle times and CPU offloading (cheaper)

Why are FPGAs/SoCs the best choice for TSN?

Overview

- **System on Chips (SoCs) (CPU and FPGA in one chip) are especially suited for the combination of TSN with OPC-UA**
  - TSN part runs in the FPGA
    - Hard real-time requirements
  - Configuration and OPC-UA runs on the CPU
    - Soft real-time requirements
  - Offloading of some functionalities from the CPU to the FPGA possible, due to tight coupling of FPGA and CPU (high bandwidth, low latency and DMA data transfer over AXI)

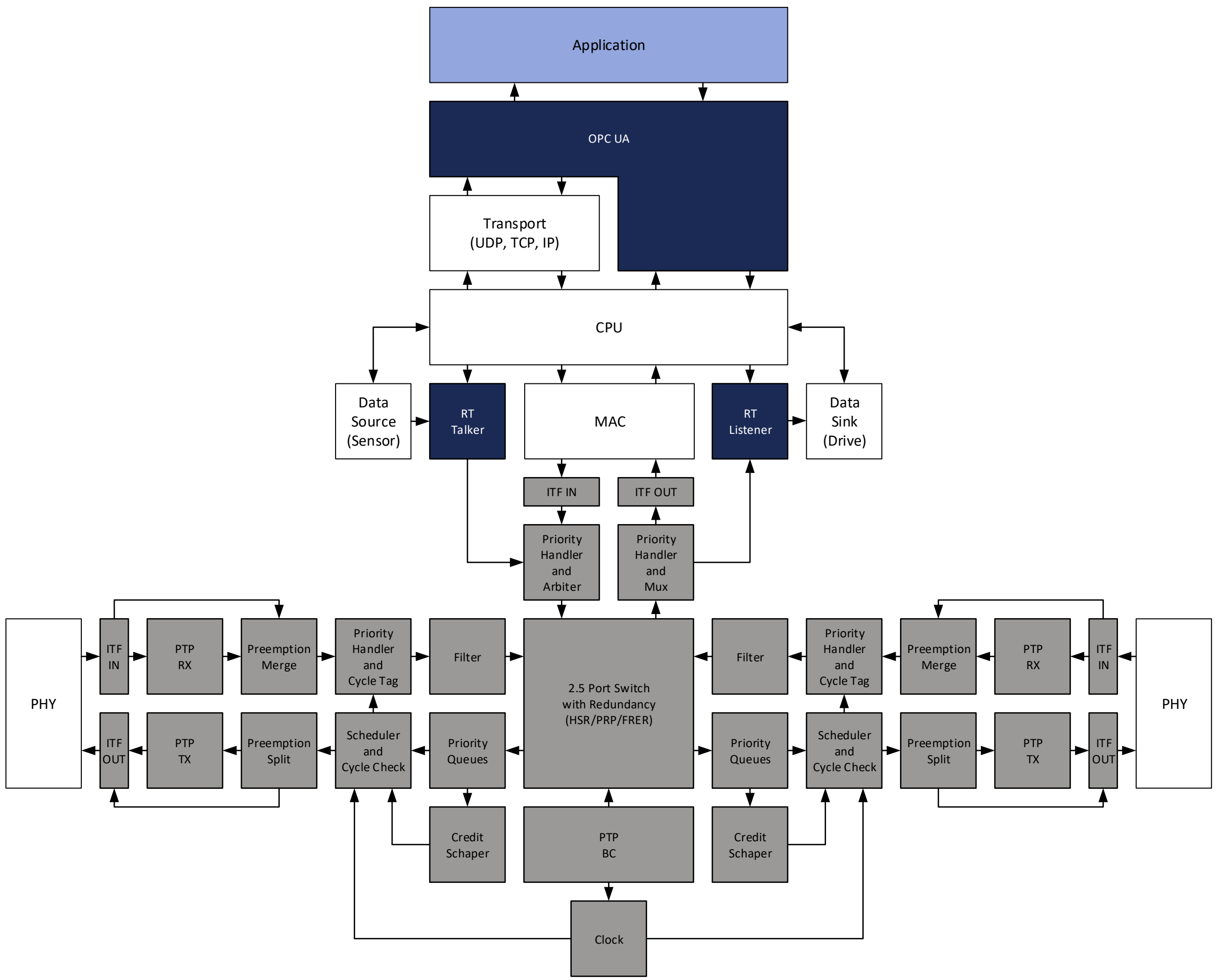
How can a TSN switched End-node be implemented using FPGA technology?

Overview

- **A switched End-node has normally 2.5 ports**
  - 2 forwarding ports
    - Either for daisy chaining, redundancy or trunk ports
  - 1 uplink port
- **Show how a switched End-node was developed in a modular manner supporting:**
  - Switching, Priority Handling, Synchronization, Scheduling, Preemption, Cyclic Forwarding, Credit Based Shaping, Redundancy and Filtering

How can a TSN switched End-node be implemented using FPGA technology?

Design



How can a TSN switched End-node be implemented using FPGA technology?

Design Advantages

- **This approach has several advantages:**
  - Step by step adding features reduced complexity
  - Clean separation of features
    - Features can be added and left away as needed
    - Clean interfaces between modules
    - Can be easily changed to an End-node (1 port)
  - Future proof for new standards/features
    - Also for more ports
  - Can be used as TSN solution with any MAC/CPU
  - Can run also as pure FPGA solution
    - Talker and Listeners in the FPGA

How can a TSN switched End-node be implemented using FPGA technology?

Design Disadvantage

- **But:**
  - To keep the design simple and to have no requirements on external RAM (DDR3/DDR4…) all FIFOs and Buffers are implemented with Block-RAM
    - The size of the FIFOs depends on the Cycle duration and Link Speed (10/100/1000Mbit/s) since in worst case a frame has to be buffered for a whole Cycle and the Link Speed defines the maximum number of frames (bytes/s)
      - For Cycle durations over 1ms and a Link Speed of >=1G it might be better to use external RAM
    - If the Cycle duration, Link Speed, Maximum Bandwidth usage and even the Schedule is known at synthesis time, buffers can be reduced quite a bit

Summary
TSN

- **Set of IEEE 802.1 standards**
  - Open standards
  - Not driven by one big company
- **Defines extensions to Ethernet**
  - Synchronization
  - Determinism and Bounded Latency
  - High Availability
- **Allows vertical integration**
  - Real-time data and best-effort data over the same communication media

Summary

Future of real-time Ethernet

- **TSN/OPC-UA will be the field communication of the future**
  - In automation, automotive, utilities etc.
  - Open (and open source OPC-UA)
- **Migration plans from existing Real Time Ethernet solutions towards TSN/OPC-UA**
- **TSN/OPC-UA is a living ecosystem which will constantly extended**
- **Many prototypes do already exist!**

Summary
FPGA and TSN

- **FPGAs and SoCs are currently the devices of choice to implement TSN/OPC-UA**
  - TSN and OPC-UA are work in progress and will be changed and extended
- **FPGAs allow offloading of Tasks from the CPU to the FPGA**
  - Short cycles and Talkers and Listeners in the FPGA
- **TSN solutions can be designed in a modular level**
  - Only instantiate what will be needed depending on the application

Thank you!

Visit us and see our IP cores and tools live

NetTimeLogic GmbH
www.nettimelogic.com
contact@nettimelogic.com