IEEE 1588 Standard for a Precision Clock Synchronization Protocol and Synchronous Ethernet

What is it?

Where is it used?

How does it work?

How to implement it?

Prof. Hans Weibel, Zurich University of Applied Sciences
hans.weibel@zhaw.ch

Who is
ZHAW – Zurich University of Applied Sciences?

Zurich University of Applied Sciences

- **ZHAW - Zurich University of Applied Sciences is a Swiss School of Engineering**
- **ZHAW‘s Institute of Embedded Systems has a strong commitment to industrial communications in general and to Ethernet and wireless, in particular, e.g.**
  - **Real-time Ethernet (Ethernet Powerlink, ProfiNet, etc.)**
  - **Time and Frequency Synchronization (IEEE 1588)**
  - **High-availability Ethernet add-ons (MRP, PRP, HSR, etc.)**
  - **Wireless (sensor networks, ultra low power Bluetooth, RFID, UWB)**
- **The related activities include**
  - **Protocol stacks**
  - **Hardware assistance and off-load (IP)**
  - **Support, engineering, and consultancy**

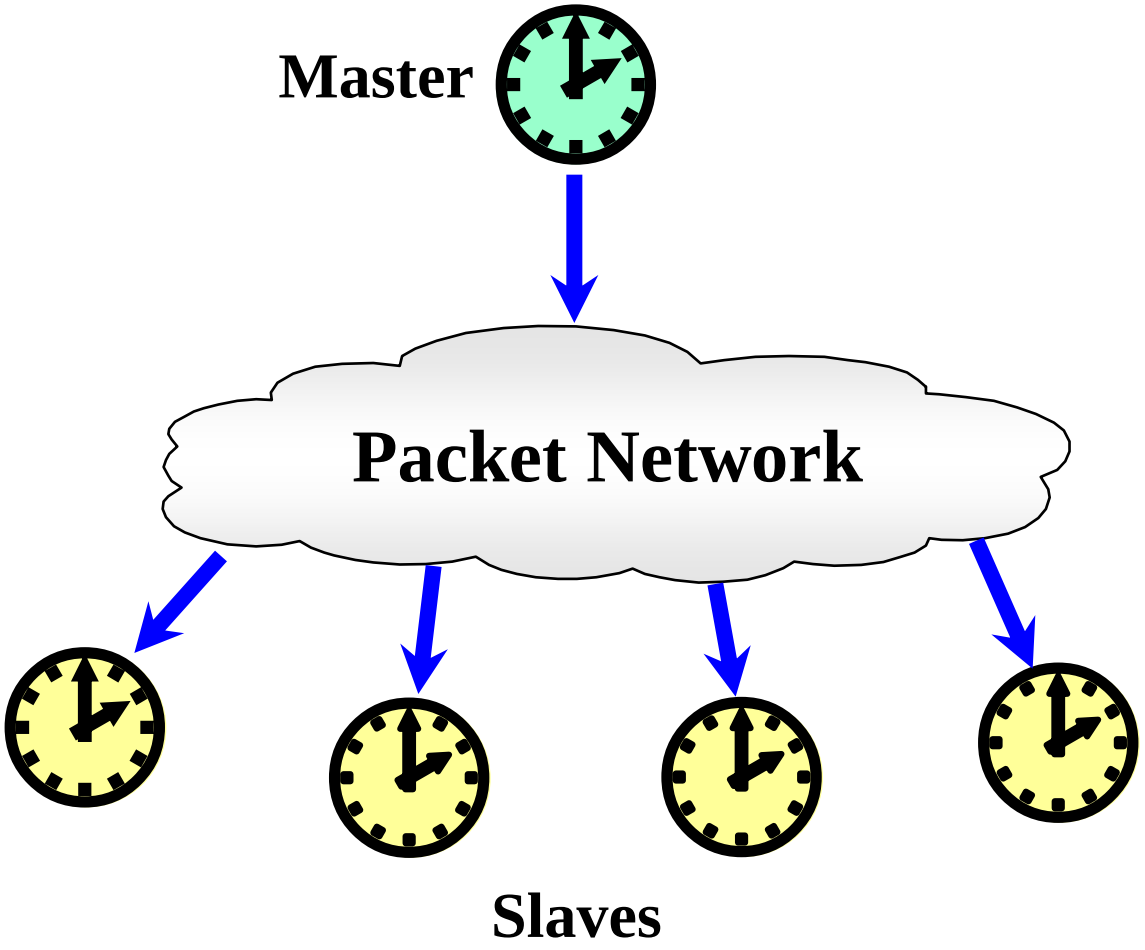
Contents

Zurich University of Applied Sciences

1) Introduction

Source: panbo.com

IEEE 1588
What is it all about?



Distribution of
- **frequency and**
- **time**

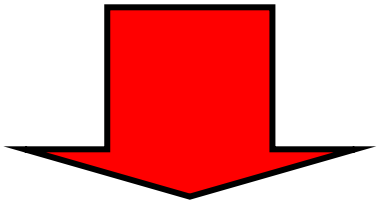
over a packet network (main focus on Ethernet)

IEEE 1588 and other Time Dissemination Networks

Why a new standard?

Zurich University of Applied Sciences

- **NTP does a good job since many years**
  - **runs on legacy data networks**
  - **but some applications demand for much higher accuracy**
- **Specialized sync networks can do this job more accurate**
  - **but at much higher cost**
  - **e.g. IRIG-B, a specialized dedicated sync network (IRIG: Inter Range Instrumentation Group)**
  - **e.g. GPS, allows for global synchronization, requires outdoor antenna**



- **IEEE 1588 offers high accuracy (< 100 ns) over a data network**
  - **but requires hardware assistance**
  - **is designed for well-controlled LAN environment**

IEEE 1588

History, Status, and Relevance

Zurich University of Applied Sciences

- **IEEE 1588 has its origins in the area of test and measurement systems.**
- **The automation and control industry has joined the group in an early stage**
- **IEEE 1588 - 2002 was approved 12th of September 2002.**
- **IEC has adopted the standard under the label IEC 61588 in 2004.**
- **New application areas generate new requirements (even higher accuracy, telecommunications industry asks for new features and characteristics).**
- **The project P1588 was started in order to specify the version 2 of the protocol.**
- **IEEE 1588 – 2008 was approved 27th of March 2008.**
- **Commercial version 2 enabled products, protocol stacks, and IPs are available.**
- **Components with integrated IEEE 1588 hardware assistance were announced or are available, e.g. microcontrollers, switching chips, and Ethernet transceivers.**
- **IEC has adopted the standard under the label IEC 61588 Ed. 2 in 2008.**
- **ZHAW's role in the area of time synchronization: It is an important R&D topic since years. ZHAW and Hirschmann Automation and Control work closely together in this area and share a common code basis and experience.**

IEEE 1588
PTP-related Standardization Activities

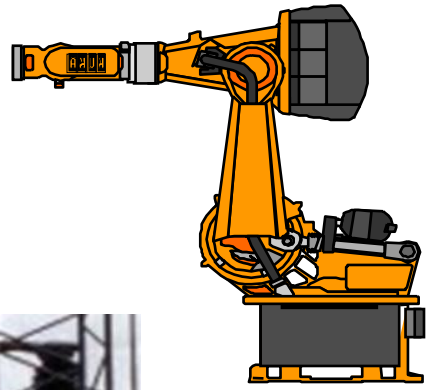
Zurich University of Applied Sciences

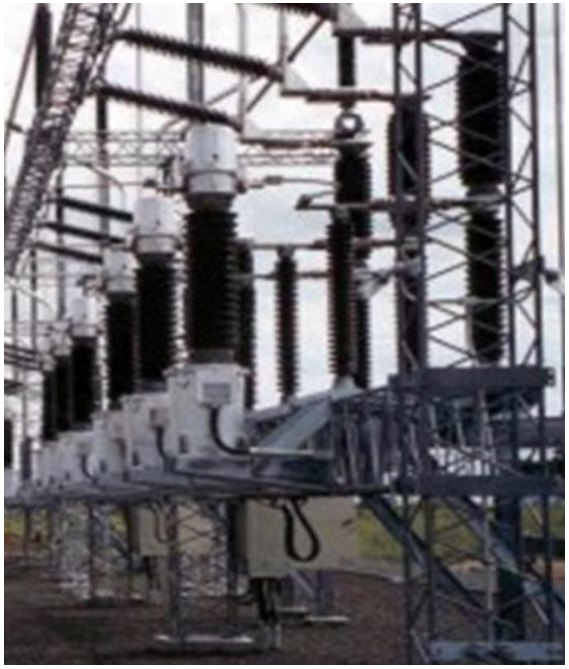
- **IEEE 1588: a small group discusses and answers interpretation problems**
- **IEEE 802**
  - **IEEE 802.1: specifies its own synchronization protocol IEEE 802.1as based on IEEE 1588**
  - **IEEE 802.3: specifies a timestamp interface IEEE 802.3bf for Ethernet PHY**
  - **IEEE 802.11: adds a timing measurement protocol to IEEE 802.11v**
- **IETF WG Timing over IP Connections and Transfer of Clock (TICTOC)**
  - **Develops PTP profile(s) for time and frequency distribution for native IP and MPLS-enabled IP networks**
- **ITU-T SG 15 / Q 13 is responsible for "network synchronization and time distribution" and is working on developing PTP profiles to address telecom applications**
- **LXI Consortium has defined a PTP profile to be used for T&M applications**
- **IEEE Power System Relaying Committee (PSRC) IEEE Standard PC37.238 titled "IEEE Standard Profile for Use of IEEE 1588 Precision Time Protocol in Power System Applications"**

2) Precision Time Protocol IEEE 1588 in General

Zurich University of Applied Sciences

2 a) Application of synchronized Clocks





System Time
Relevancy of a common Time Base

Zurich University of Applied Sciences

zhaw

System time helps

- to coordinate measurement instants (sampling, triggering)
- to measure time intervals (and to calculate derived quantities)
- as a reference to determine the order of events
- to determine the age of data items (data correlation; data base replication)
- as a basis for the execution of coordinated actions (time based behaviour)
  - scheduled execution of scripts
  - scheduled execution of mutual exclusion
- to generate frequencies
- to decouple communication from execution

The system wide provision of „exact“ system time offers new approaches to implement distributed measurement and control systems.

Application of synchronized Clocks

Where is sub-µs Accuracy required?

Zurich University of Applied Sciences

- **Automation and control systems**
  - Synchronize multi axis drive systems
  - Synchronize subsystems with cyclic operation
- **Measurement and automatic test systems**
  - Correlation of decentrally acquired values
  - Time stamping of logged data
- **Power generation, transmission and distribution systems**
  - Control of switching operations
  - Reconstruction of network activities and events
  - Isolation of problems (distinguish cause and impact)
- **Ranging, telemetry and navigation**
  - Triangulation
  - Large sensors for seismic or submarine applications
- **Telecommunications and consumer electronics**
  - Distribution of frequency and time in Next Generation Networks
  - Emulation of TDM circuits through packet networks
  - Synchronization of wireless base stations
  - Backup for other time sources (loss of GPS signal)
  - Audio/video transmission

Application of synchronized Clocks
Automation and Control Systems

Zurich University of Applied Sciences

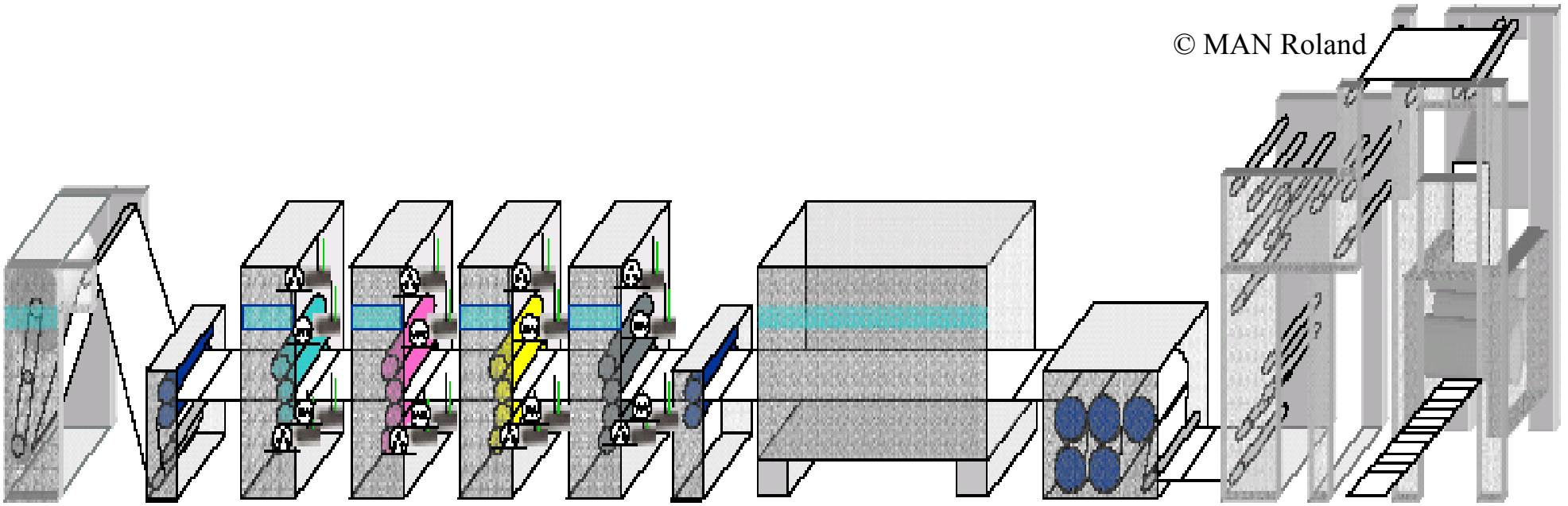
zh aw

Example: Multi axis motion control, e.g. in a printing machine

- **Many drives have to be synchronized**

Application of synchronized Clocks

Offset Printing Machine



speed up to v = 20 m/s

printing accuracy Δs = ± 5 µm

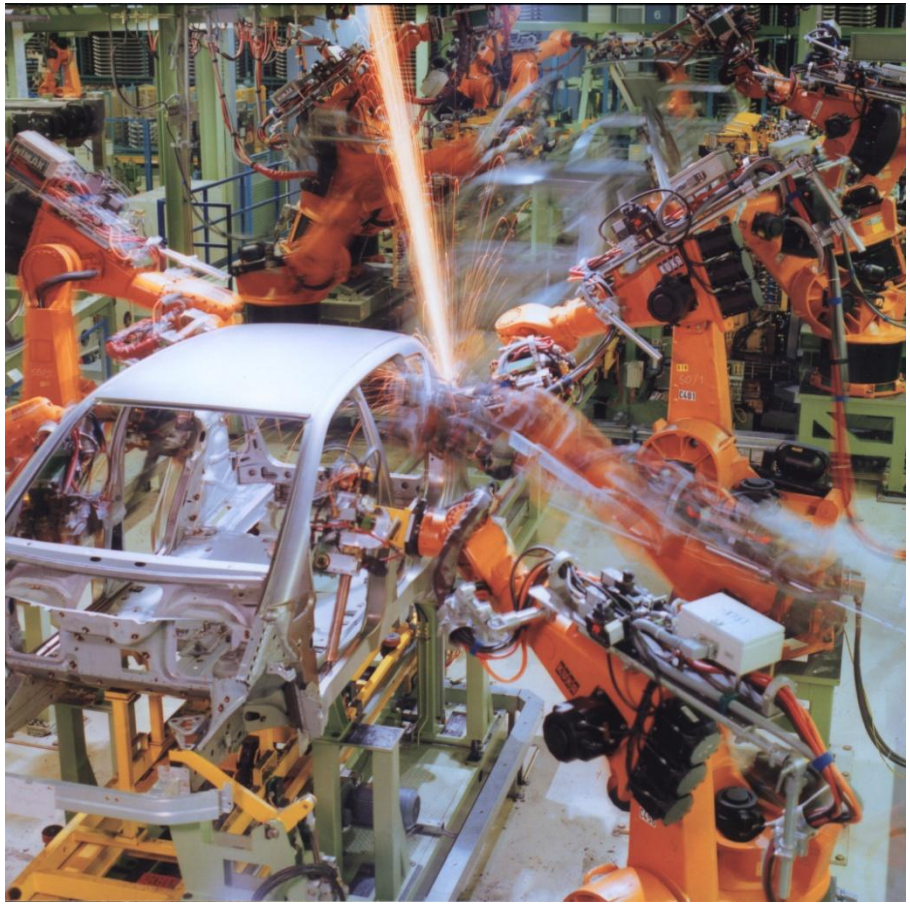
→ synchronization requirement: Δs/v = ± 250 ns

Application of synchronized Clocks

Automation and Control Systems

Zurich University of Applied Sciences

zh aw

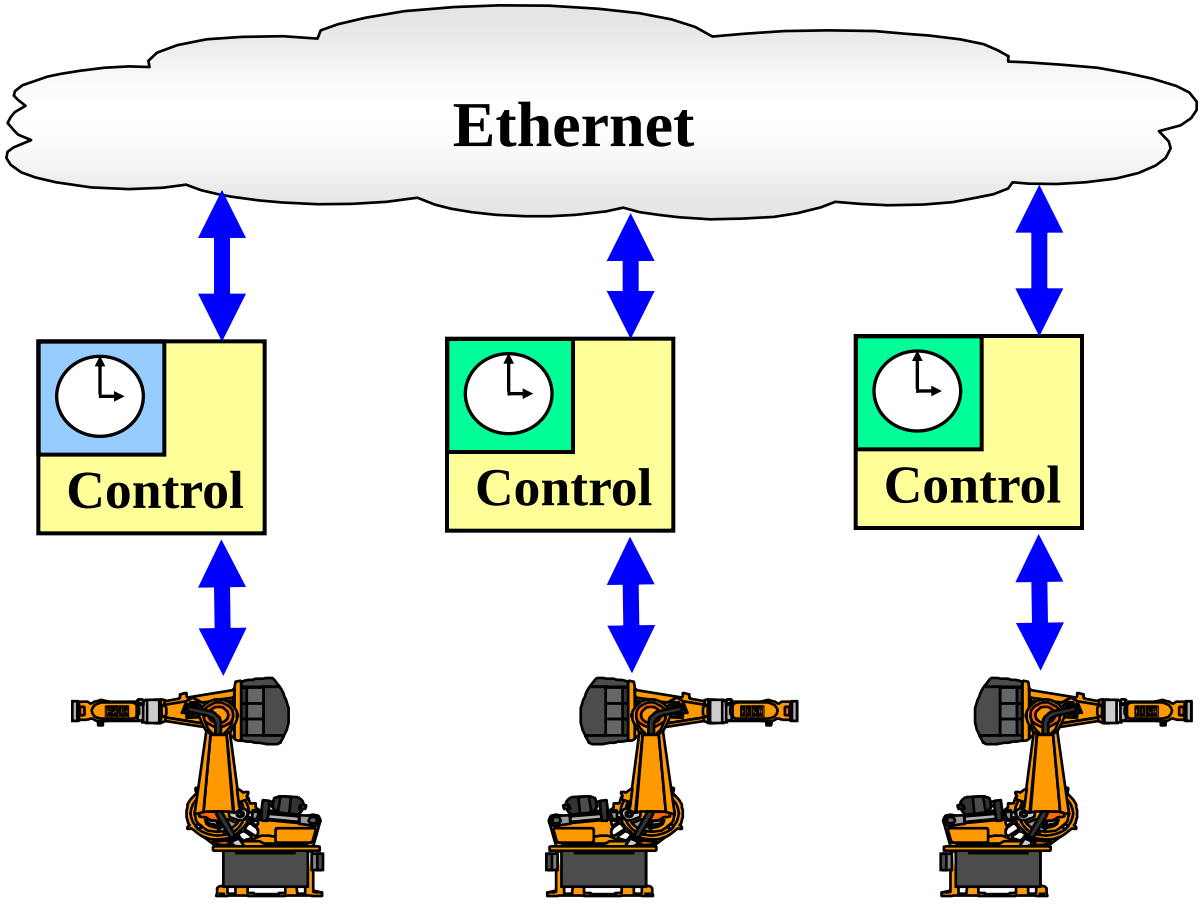


© Picture: KUKA Roboter GmbH

Example: Multi axis motion control, e.g. in coordinated robots

Application of synchronized Clocks

Cooperating Roboters (RoboTeam)



Application of synchronized Clocks

RoboTeam in action: Load Sharing

Application of synchronized Clocks
Measurement, Data Acquisition, and Test Systems

Zurich University of Applied Sciences

- **Capture / acquire data**
  - **within a distributed environment**
  - **simultaneously at different places**
  - **deliver data with a time stamp**
- **Processing**
  - **Correlate the data**
  - **Report the order of events**
  - **Reconstruct complex, fast and distributed activities**
- **Straight forward installation**
  - **Sync and data over the same network**
  - **no special sync lines required**

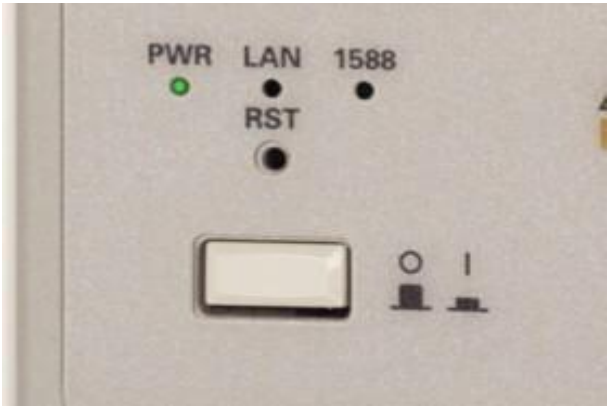
Application of synchronized Clocks

T&M - LAN eXtensions for Instrumentation (LXI)

- **Initiative pushing Ethernet to interconnect instruments (LAN as a replacement of IEEE 488, also known as GPIB or HPIB)**
- **Common LAN implementation**
- **Small modules and traditional box instruments**
- **Web browser interface for set-up, control, and data access**
- **Different trigger options**
  - **Hardware trigger**
  - **Software triggering over LAN using IEEE 1588 Precision Time Protocol → simplifies cabling**
- **Peer-to-peer communication between devices reduces controller traffic and can speed test throughput**
- **Common sense of time simplifies instrument synchronization**

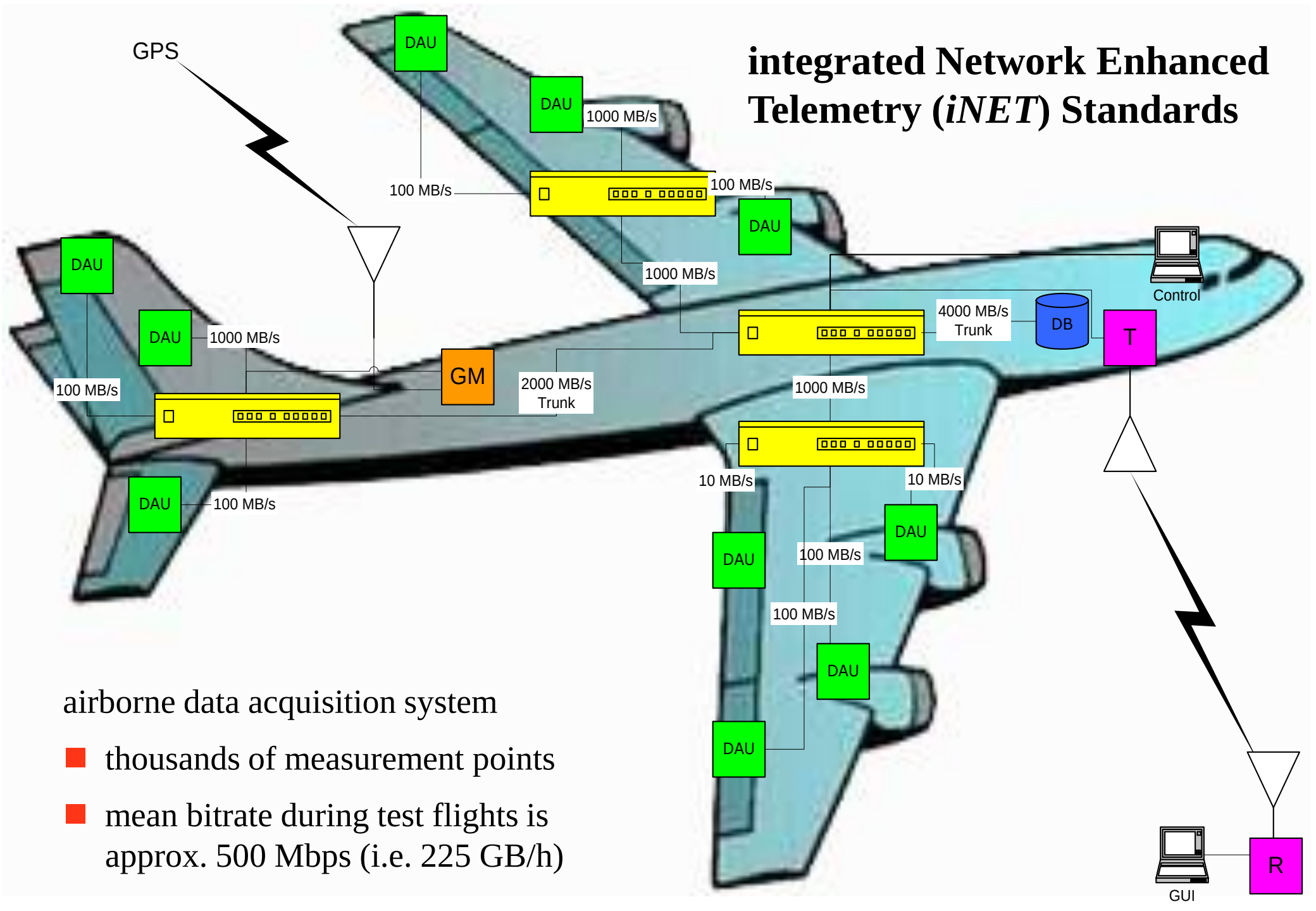
Application of synchronized Clocks

LXI Equipment



Data Acquisition for Aerospace Applications

integrated Network Enhanced Telemetry (*iNET*) Standards



airborne data acquisition system

- thousands of measurement points
- mean bitrate during test flights is approx. 500 Mbps (i.e. 225 GB/h)

Data acquisition in the old days: 707 test flight

(Poster at the Air and Space museum in Washington)

Application of synchronized Clocks

Power Generation, Transmission and Distribution

Zurich University of Applied Sciences

zh aw

- Components of the power grid have to be protected from critical load situations and turned off
- Protection switching guarantees high service availability
- U/I is measured at all critical points within the grid at precise time points and in a high rate
  - to monitor the network
  - to predict critical load situations
  - to protect the network from overload
  - to measure delivered/consumed power between providers
- The traditional solution for synchronization is based on a dedicated and costly cabling
- Using the same network for data and synchronization has big economic and operational advantages

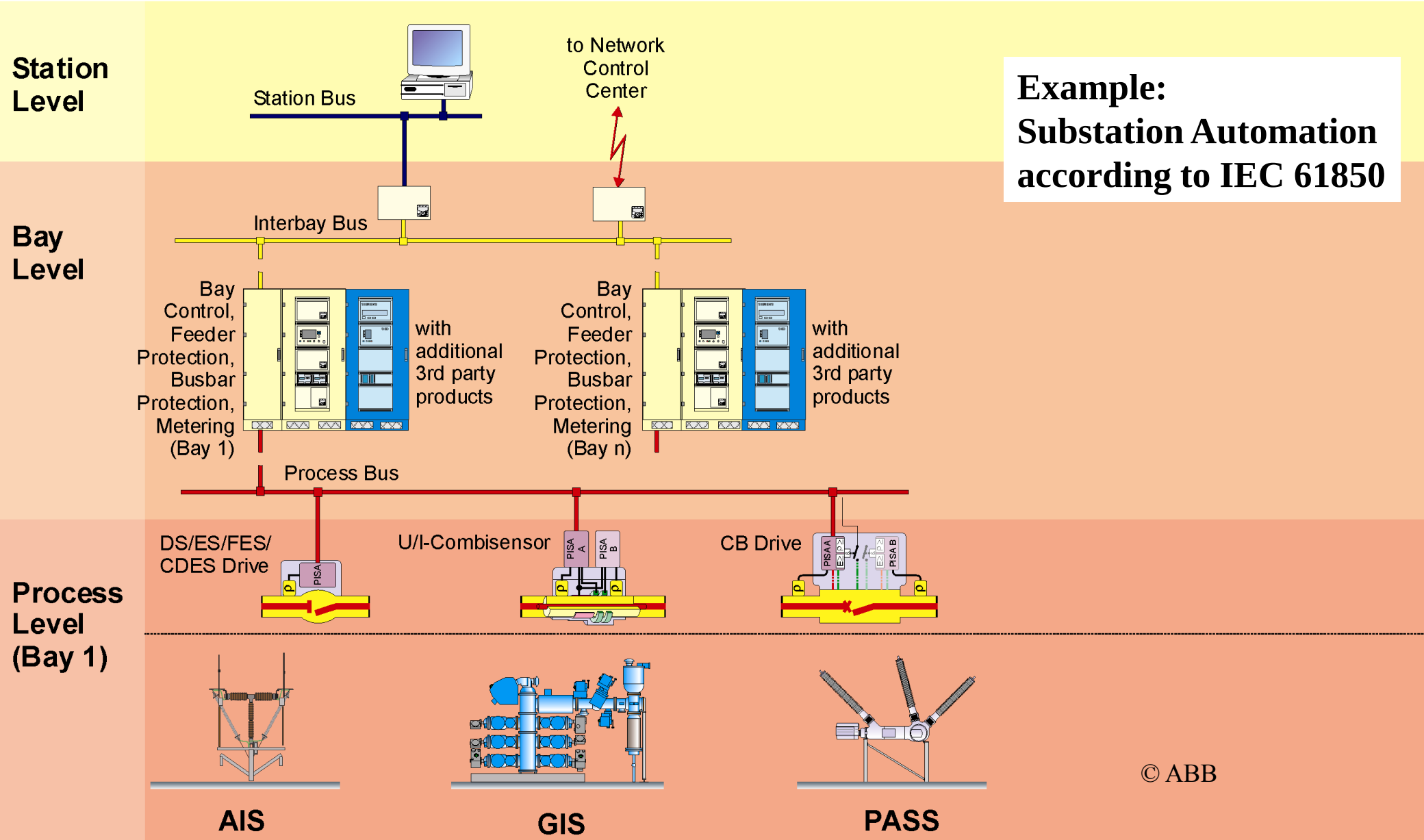
Application of synchronized Clocks

Substation Automation (based on IEC 61850)

© ABB

Application of synchronized Clocks

Substation Control and Protection



Example: Substation Automation according to IEC 61850

Application of synchronized Clocks
Synchronization Requirements

Zurich University of Applied Sciences

zh aw

Today's Requirements:

- **Most devices to be synchronized to an accuracy of 1 millisecond today.**
- **Some special applications require 100 and down to 35 microseconds today.**
- **Trend towards distributed systems and a general requirement on 35 microseconds.**

New Technologies and Applications are more demanding:

- **New technologies like Non Conventional Instrument Transformers (NCITs) for GIS require a synchronization of all or most devices to an accuracy of 1 to 4 microseconds.**
- **Target is therefore 1 microsecond for all systems with these technologies and applications.**
- **The introduction of the above mentioned technologies will take years, but also present applications will benefit from a higher accuracy.**

Application of synchronized Clocks
Communications

Zurich University of Applied Sciences

- **Frequency distribution over packet networks**
  - **Circuit Emulation Service in packet networks (TDM over Packet) → becomes more and more attractive in IP-centric infrastructure**
  - **compelling solution in pure Ethernet configurations such as Metro Ethernet or Ethernet in the First Mile (EFM)**
- **Base station synchronization**
  - **Mobile network backhaul (e.g. femto cell synchronization)**
  - **Single Frequency Networks (SFN) such as Trunk Radio systems, DVB, and LTE MBMS (Multimedia Broadcast Multicast Services), where all transmitters are synchronously modulated with the same signal and operate on the same frequency**
- **QoS monitoring (one-way delay measurement), SLA-compliance checking (see IETF MPLS WG draft „Packet Loss and Delay Measurement for MPLS Networks") and billing mechanisms**
- **Audio/Video Bridging (AVB) according to IEEE 802.1as, 802.1Qat, 802.1Qav**
  - **Synchronization allows the transmission of very low-latency audio and video streams**
  - **no need to fill a large buffer before stream can be played**

Application of synchronized Clocks

Requirements at the Air Interface of Wireless Networks

Zurich University of Applied Sciences

Variations in the radio frequency of cellular base stations affect the ability of the system

- **to hand-over calls without interruption**
- **to minimize interference**

Mobility Standard	Frequency	Time/Phase
CDMA2000	50 ppb	Range: <3µs to <10µs
GSM	50 ppb	
WCDMA	50 ppb	
TD-SCDMA	50 ppb	3µs inter-cell phase Δ
LTE (FDD)	50 ppb	
LTE (TDD)	50 ppb	*3µs inter-cell phase Δ
LTE MBMS	50 ppb	*5µs inter-cell phase Δ
WiMAX (TDD)	50 ppb inter BTS	Typically 1 - 1.5 µs
Backhaul	1 to 16 ppb	

* Standards being consolidated

50 ppb or 5×10^{-8}

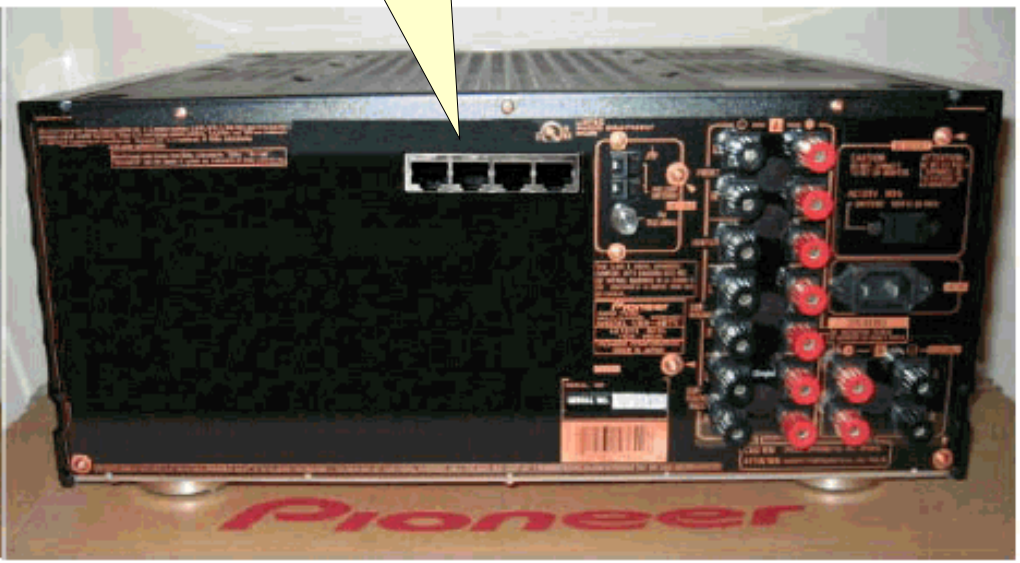
Source: Symmetricom

Application of synchronized Clocks
Audio/Video Bridging (AVB)

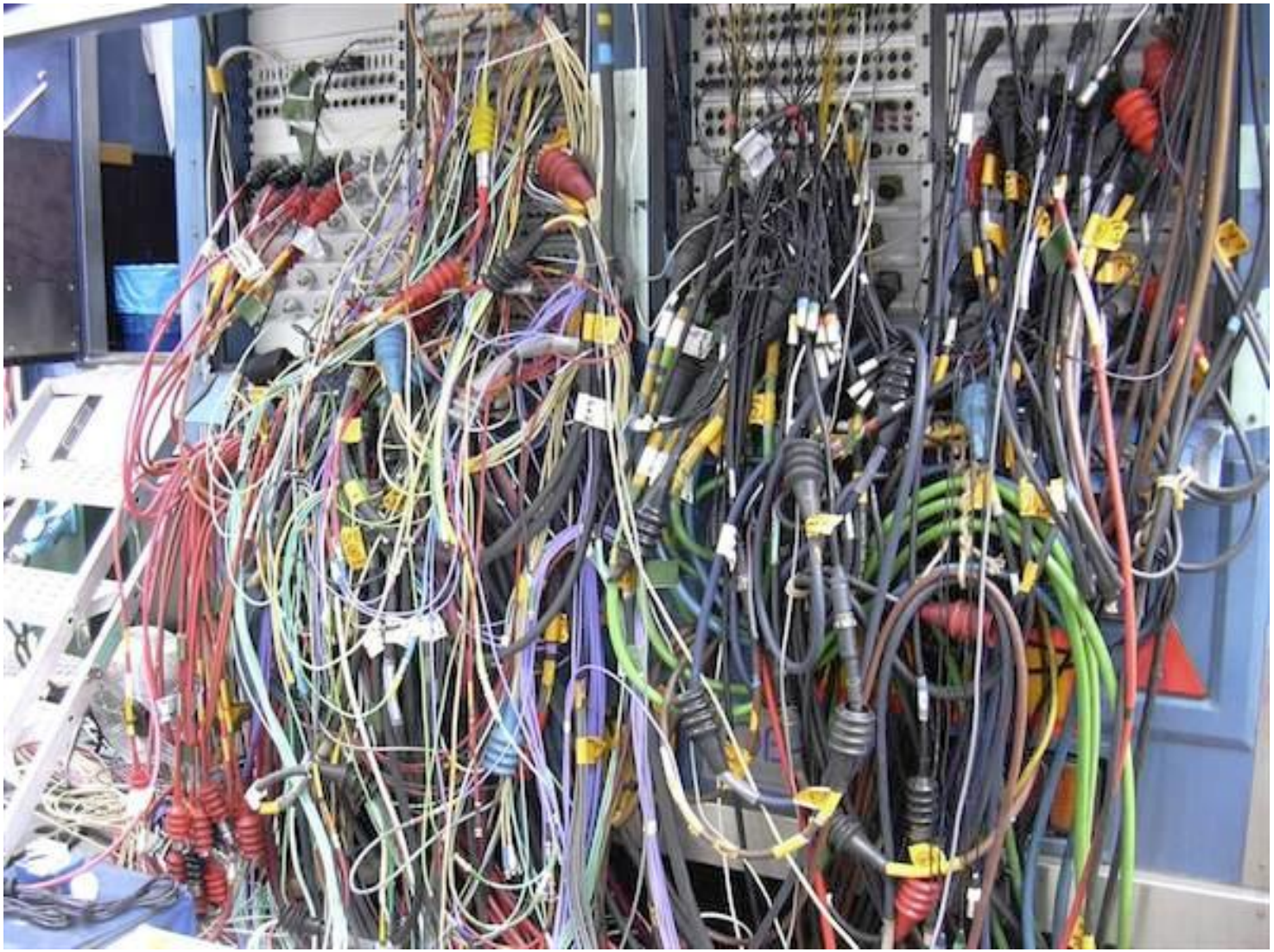
Zurich University of Applied Sciences

zh aw

The next generation audio/video connector



Application of synchronized Clocks
Audio/Video Bridging (AVB)



Studio Wiring

Source:
Wikipedia

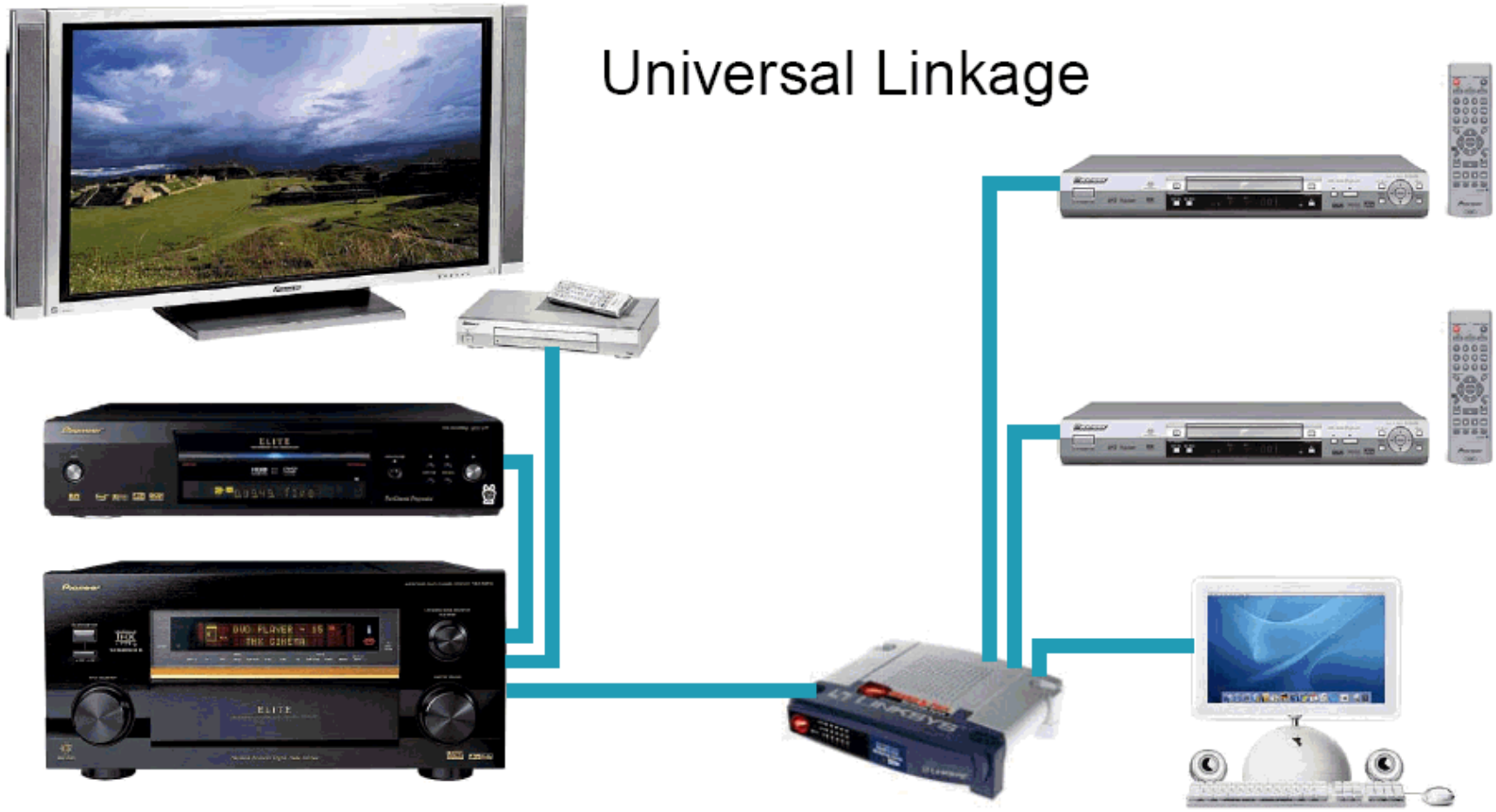
Application of synchronized Clocks
Audio/Video Bridging (AVB)

Zurich University of Applied Sciences

- **IEEE standardization effort**
  - **Allow low-latency streaming services through 802 networks**
  - **Provide end-to-end Quality-of-Service (i.e. low and guaranteed latency)**
- **AVB synchronization is specified for 802.3 full-duplex, 802.3 EPON, 802.11, and MoCA (Multimedia over Coax Alliance)**
- **Important new mechanisms**
  - **IEEE 802.1as defines synchronization (subset of IEEE 1588 with minor modifications)**
  - **IEEE 802.1Qat specifies a Stream Reservation Protocol (SRP)**
  - **IEEE 802.1Qav specifies Forwarding and Queuing Enhancements for Time-Sensitive Streams**
- **Addressed applications are e.g.**
  - **Home communication and entertainmen systems, e.g. multiple high definition video streams over an Ethernet link or Ethernet connected speakers**
  - **Car communication and entertainmen systems**
  - **Professional audio/video studio equipment**

Application of synchronized Clocks

Audio/Video Bridging (AVB)

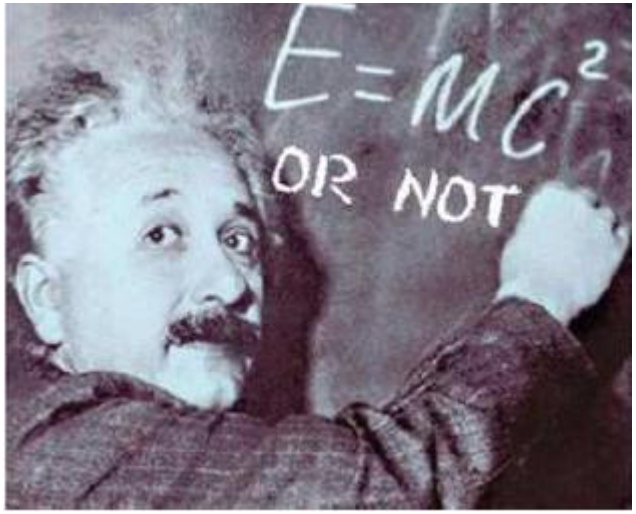


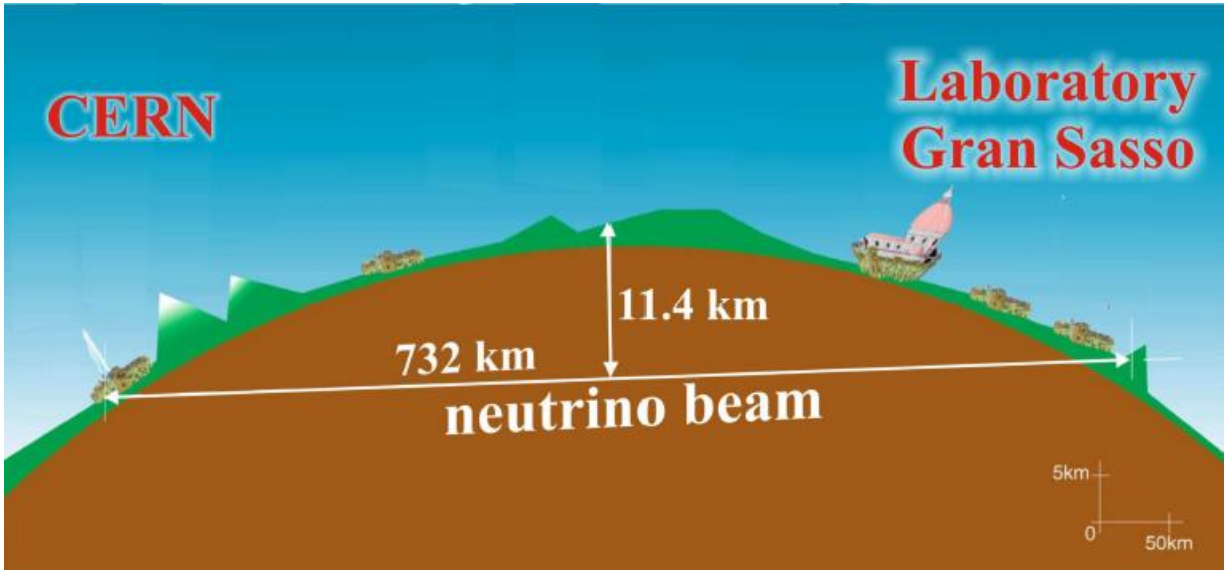
Application of synchronized Clocks
Experimental Physics

Zurich University of Applied Sciences

Special measurement and control applications

- **e.g. accelerator control and timing**
- **or Time of Flight measurement of neutrinos ☺**





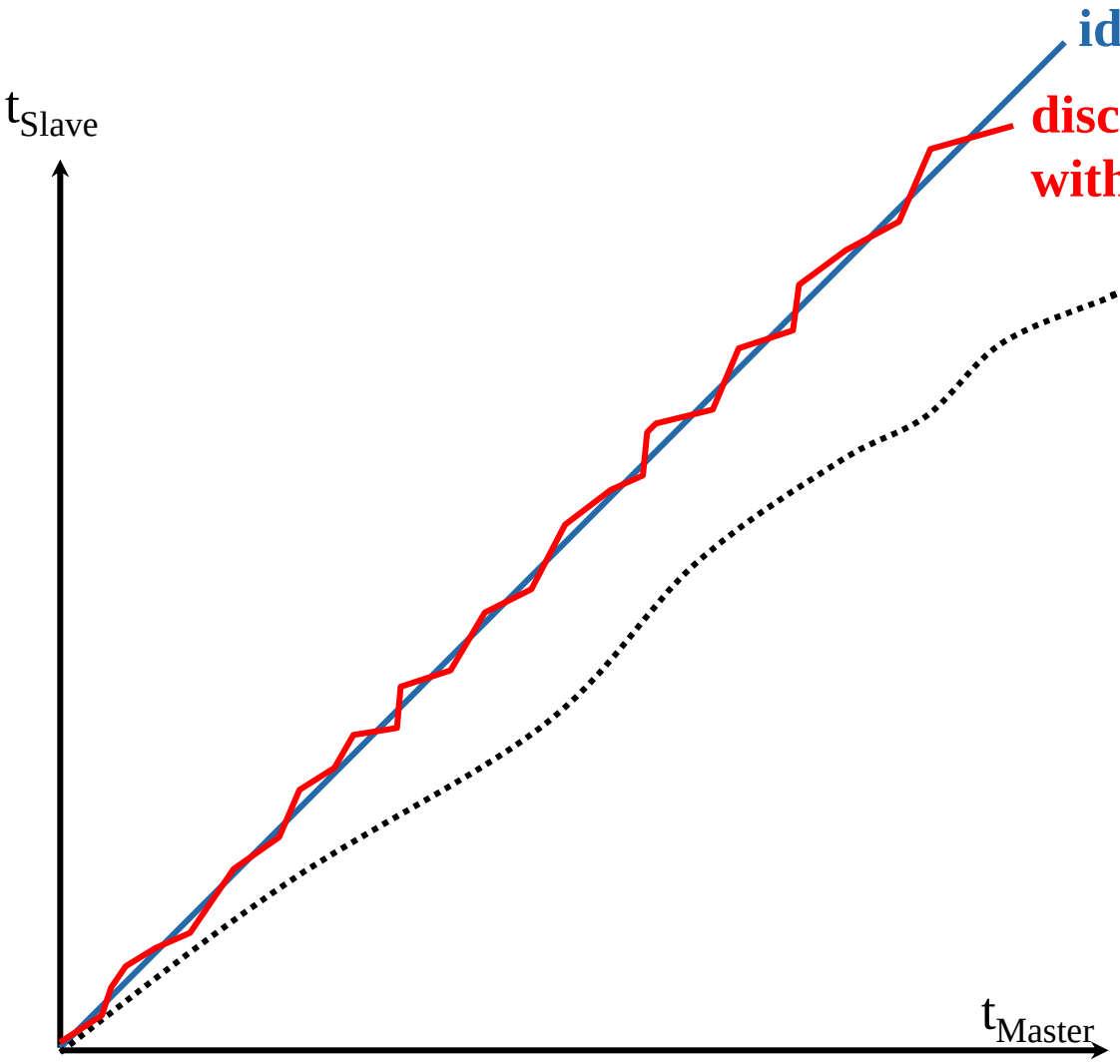
Source: Maciej Lipinski, CERN

Zurich University of Applied Sciences

2 b) Basic Synchronization Operation

Basic Operation of IEEE 1588

Effect and Limitations of Clock Adjustment

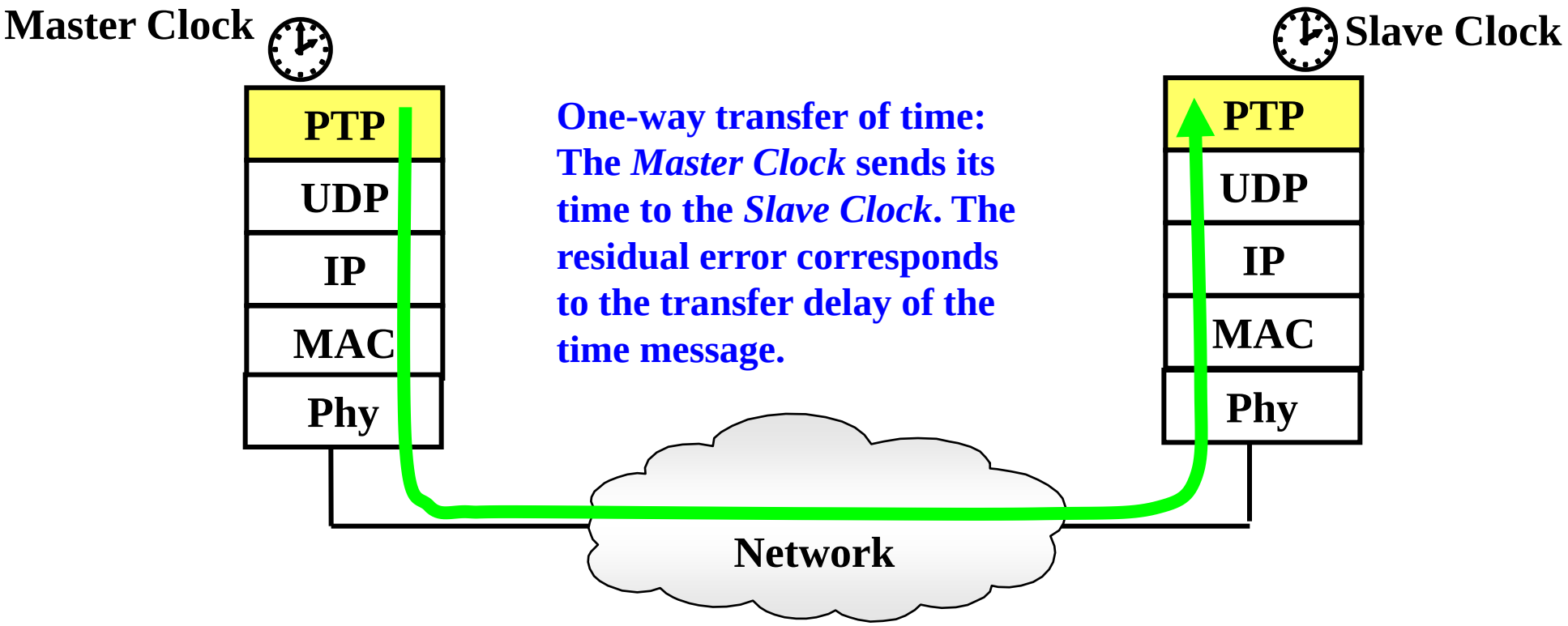


free running : the slave's oscillator

- **is in this example on average a little bit too slow (difference between nominal and effective frequency)**
- **its drift (phase change rate) varies with time (subject to temperature, acceleration, ageing, etc.)**

Basic Operation of IEEE 1588

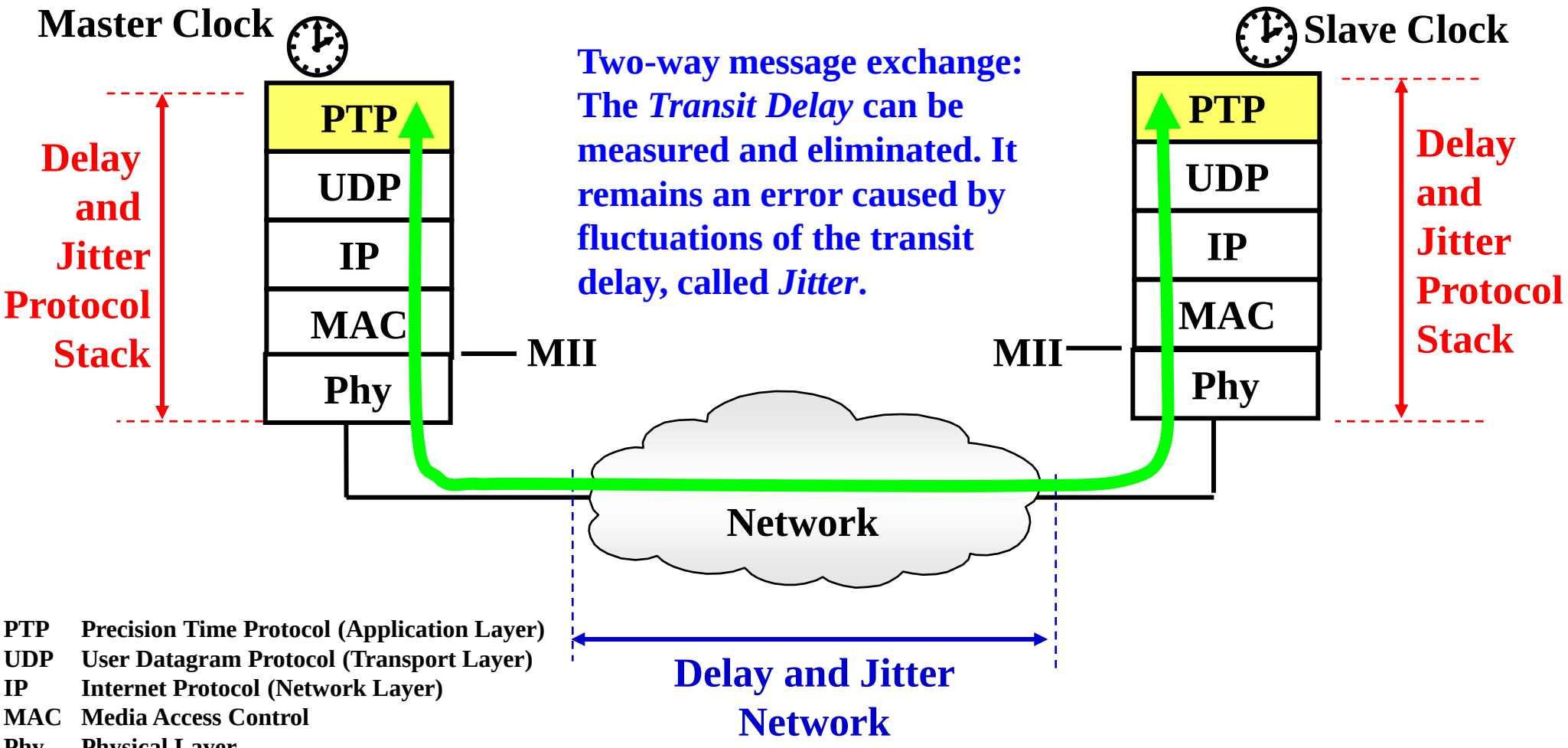
Principle of Clock Adjustment



PTP Precision Time Protocol (Application Layer)
UDP User Datagram Protocol (Transport Layer)
IP Internet Protocol (Network Layer)
MAC Media Access Control
Phy Physical Layer

Basic Operation of IEEE 1588

Transit Delay and Jitter



PTP	Precision Time Protocol (Application Layer)
UDP	User Datagram Protocol (Transport Layer)
IP	Internet Protocol (Network Layer)
MAC	Media Access Control
Phy	Physical Layer

Basic Operation of IEEE 1588
Elements of the Solution

Zurich University of Applied Sciences

zhaw

Delay and jitter caused by the protocol stack

- **Measurement messages are generated and processed on the application layer, but the respective timestamps are taken as near as possible to the wire**
  - **i.e. somewhere on the physical layer**

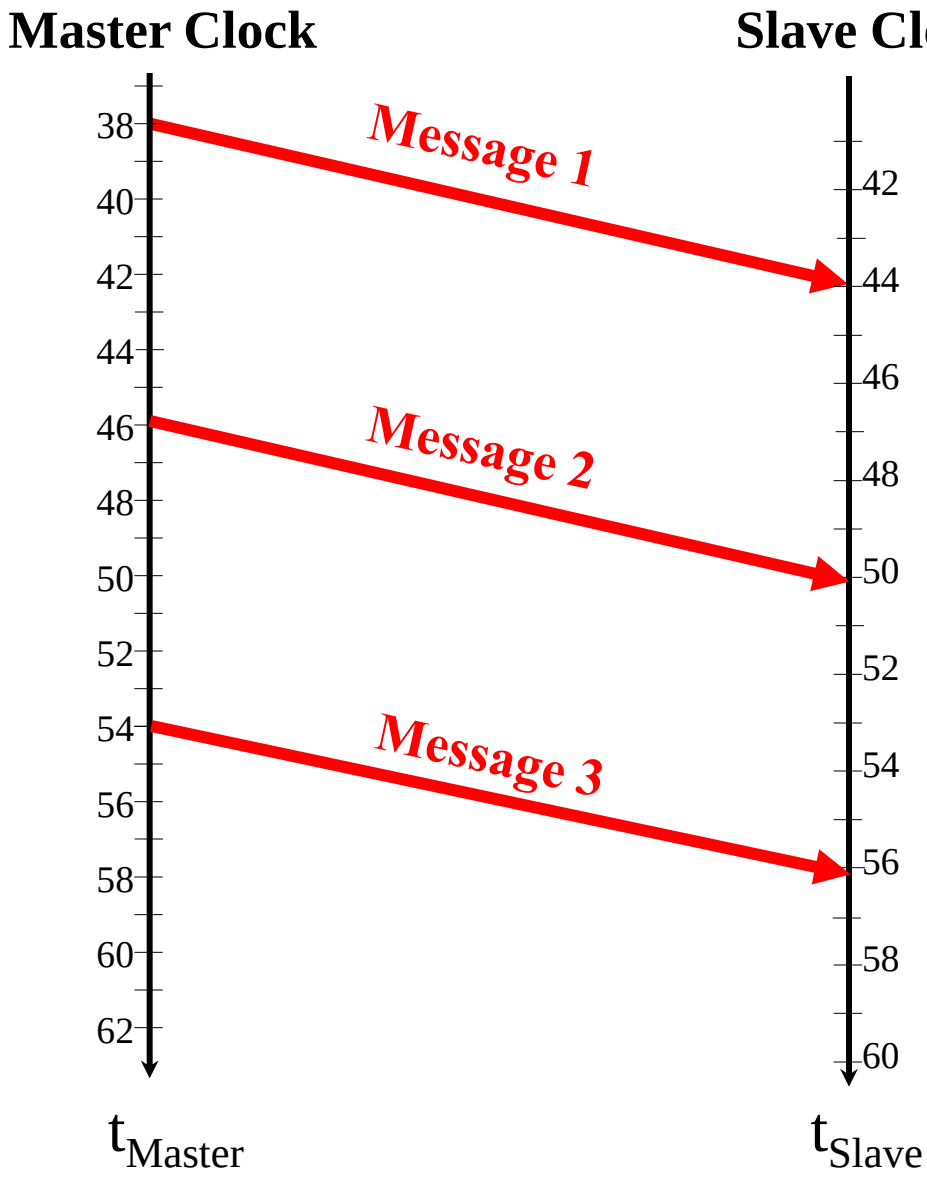
Jitter caused by the network

- **Specify and use IEEE 1588 – aware network infrastructure which is capable to handle this problem**
  - **i.e. use special switches such as Boundary Clocks (BCs) and/or Transparent Clocks (TCs)**
- **or use legacy infrastructure and use statistical methods**

As a didactic step, we start to understand PTP's operation by using a network with a constant transit delay (e.g. a cable)

Basic Operation of IEEE 1588

Two different Clocks are used to measure Transit Delay



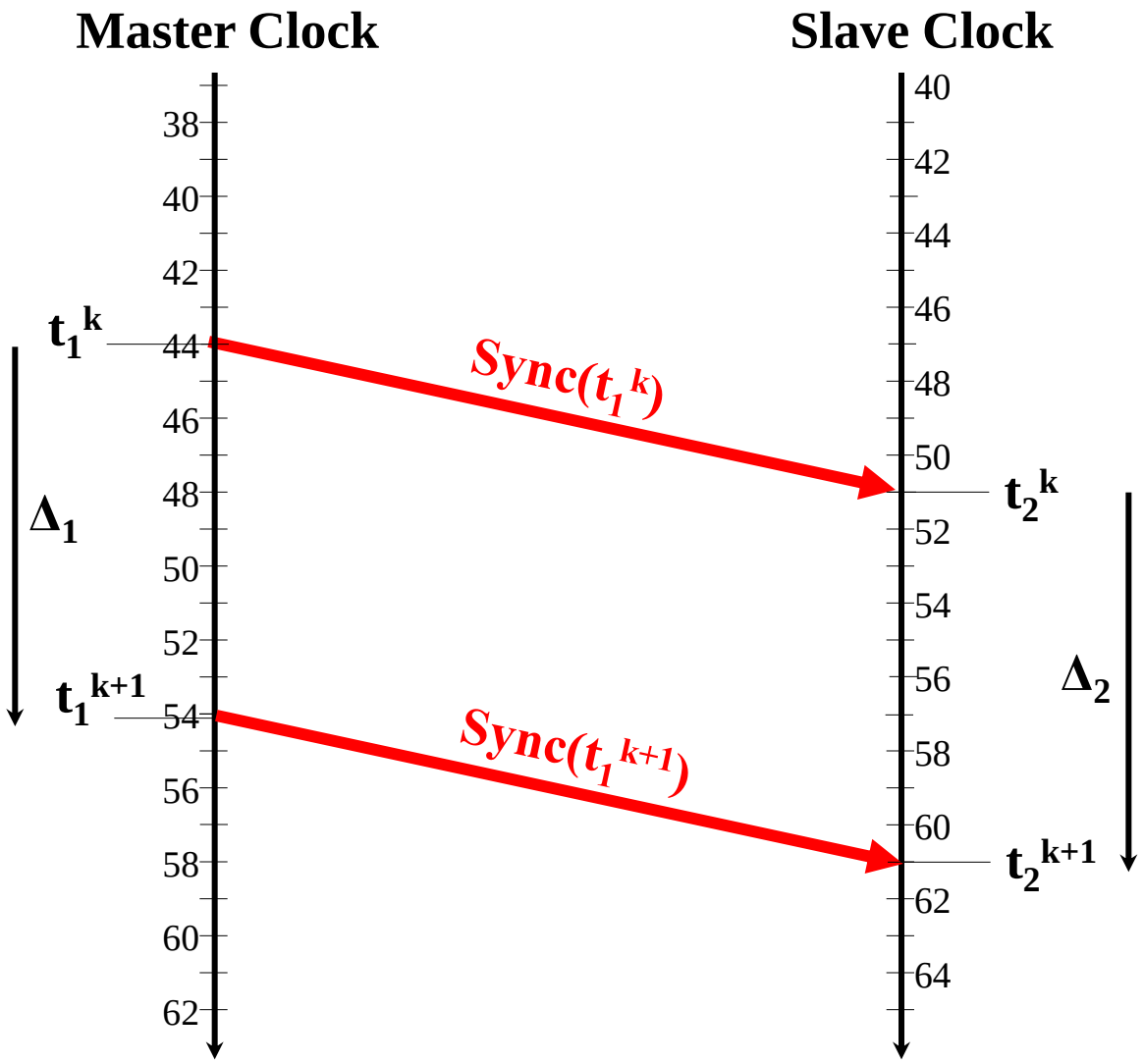
$t_{Slave} - t_{Master} = 6$

$t_{Slave} - t_{Master} = 4$

$t_{Slave} - t_{Master} = 2$

Basic Operation of IEEE 1588

Determination of Phase Change Rate (Drift) – one-step clock



One-step clock:
Special hardware required in order to insert timestamp t_1 on the fly

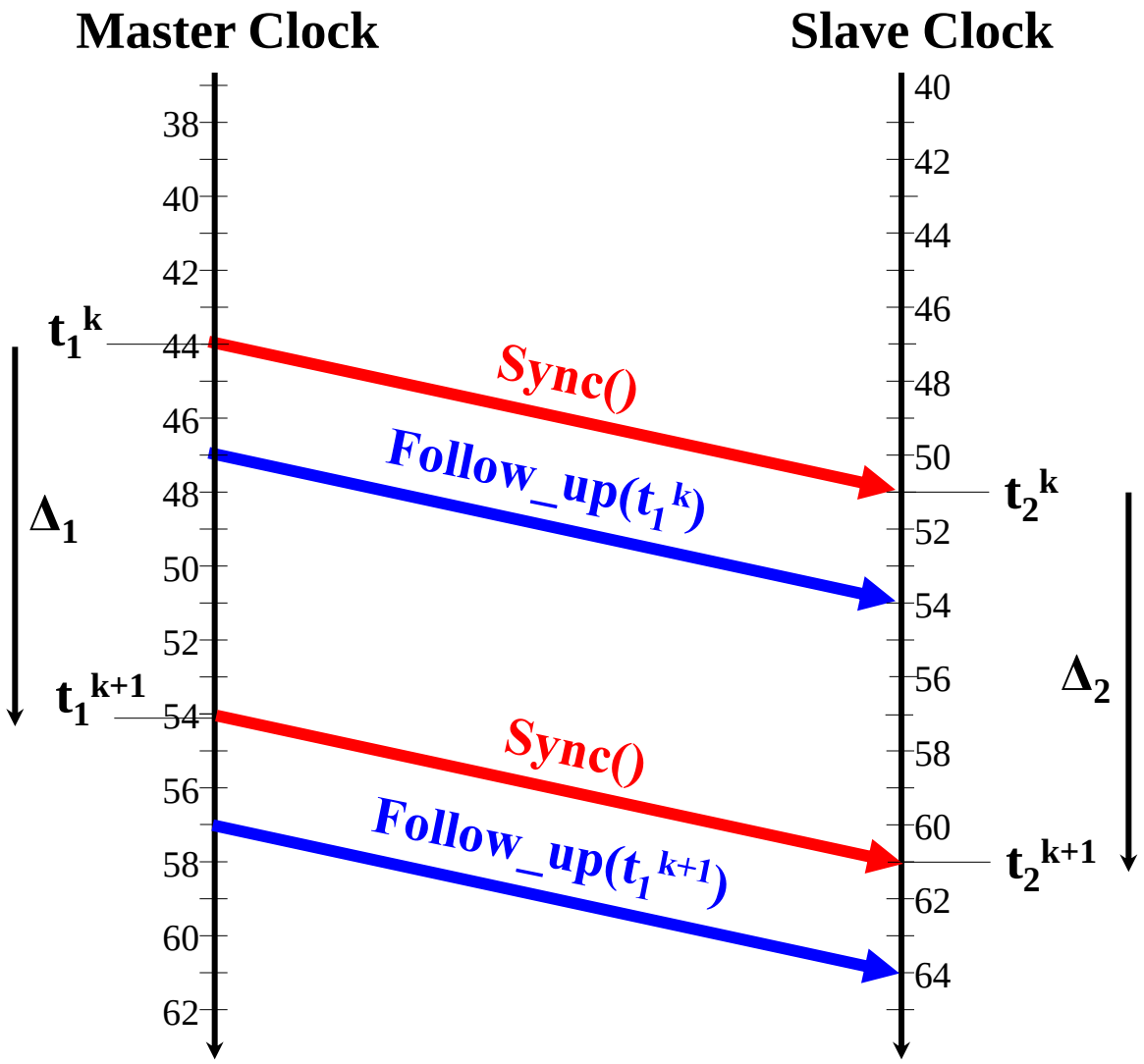
$$\Delta_1 = t_1^{k+1} - t_1^k$$

$$\Delta_2 = t_2^{k+1} - t_2^k$$

$$\text{Drift} = \frac{\Delta_2 - \Delta_1}{\Delta_2}$$

Basic Operation of IEEE 1588

Determination of Phase Change Rate (Drift) – two-step clock



Two-step clock:
No need to modify Sync message on the fly

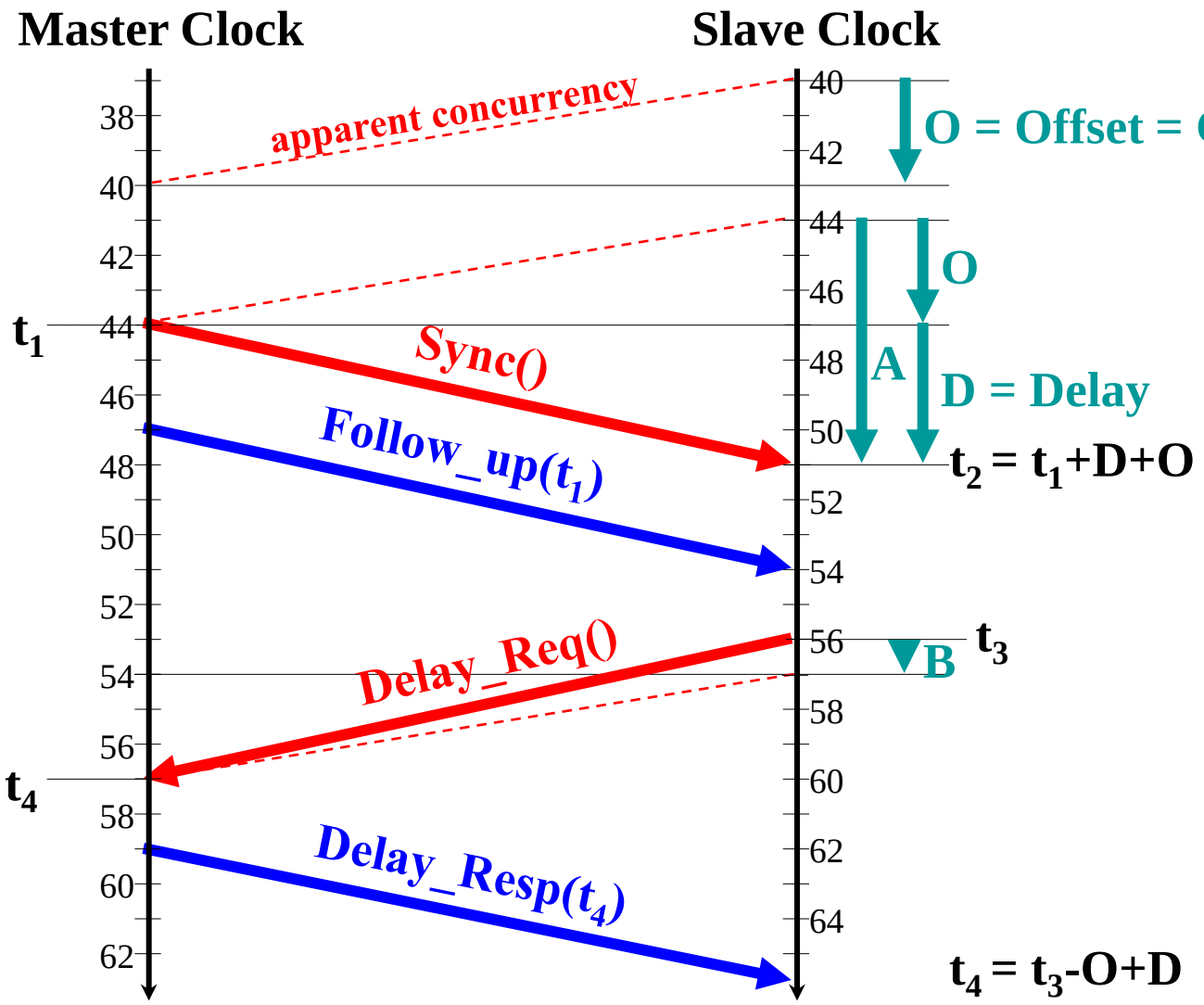
$$\Delta_1 = t_1^{k+1} - t_1^{k}$$

$$\Delta_2 = t_2^{k+1} - t_2^{k}$$

$$\text{Drift} = \frac{\Delta_2 - \Delta_1}{\Delta_2}$$

Basic Operation of IEEE 1588

Delay and Offset Determination



measured values t_1, t_2, t_3, t_4

$A = t_2-t_1 = D+O$

$B = t_4-t_3 = D-O$

$$\text{Delay } D = \frac{A + B}{2}$$

$$\text{Offset } O = \frac{A - B}{2}$$

Basic Operation of IEEE 1588

Accuracy

Zurich University of Applied Sciences

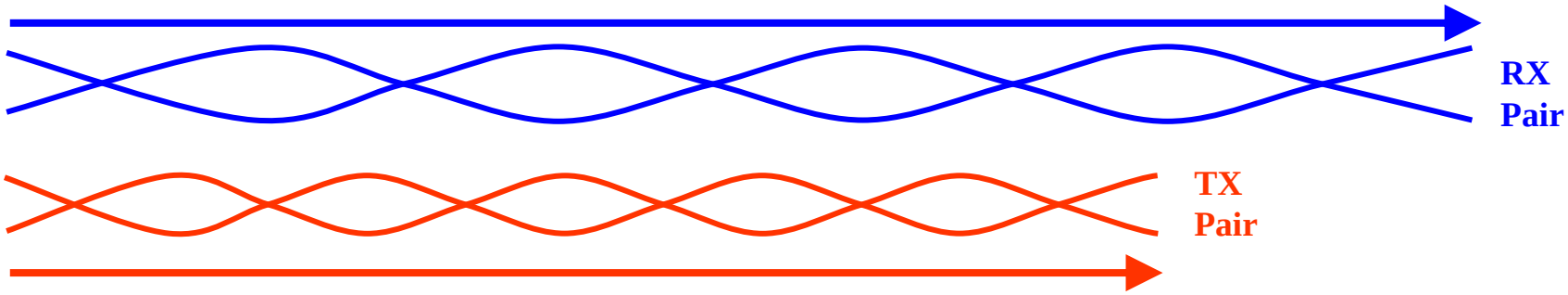
zh aw

- **Fundamental assumption:**
  The one-way transmission delay is estimated being half of the round trip time
- **Accuracy depends on two major factors**

→ Symmetry of the transmission path

→ Accuracy of timestamps

Basic Operation of IEEE 1588
Example of a Medium Asymmetry



- Different twist rate of twisted line pairs leads to delay skew (difference between propagation delays of transmit and receive lines)
  - CAT 5/6: allows up to 50 ns per 100 meter cable (IEC 11801)
  - CAT 7: allows up to 30 ns per 100 meter cable (IEC 11801)
  - Real cables are typically better than IEC 11801 allows
  - PROFINET quad-cable: allows 8 ns per 100 meter cable
- Very high symmetry can be achieved with single fiber operation (two different wavelength on the same fiber)

Basic Operation of IEEE 1588
Frequency and Time Transfer

Zurich University of Applied Sciences

- **Phase Change Rate (Drift) Compensation → Frequency Transfer**
  - The slave‘s oscillator does not have exactly the same frequency as the master
  - The slave‘s oscillator frequency varies over time (due to environmental conditions)
  - Consecutive timestamped Sync messages allow to determine and compensate the deviation (accelerate or slow down the oscillator)
  - This compensation is repeated regularly (frequency depending on oscillator stability and desired accuracy)
  - A frequency aligned slave clock is called "syntonized„
  - Syntonization can be accomplished with one-way communication
  - Remember: 1 ppm is 1 µs / s
- **Offset Correction → Time Transfer**
  - Set the slave‘s time (of day) to the master‘s time
  - Correction is based on the round trip time measurement (carried out by timestamped Sync and Delay_Req messages)

Basic Operation of IEEE 1588
Offset Computation and Correction

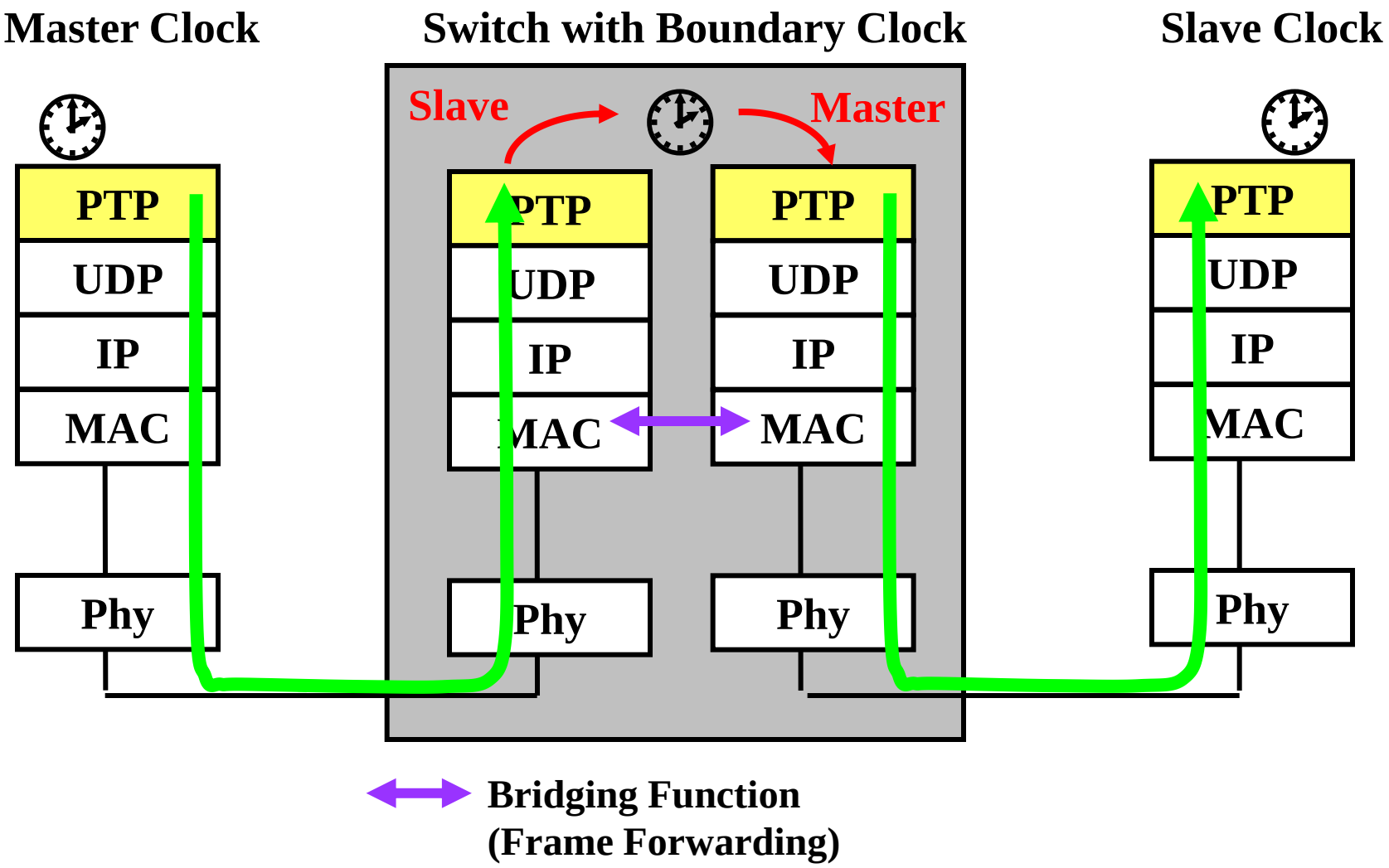
Zurich University of Applied Sciences

- The standard defines how master and slaves communicate and where timestamps are generated
- The standard says nothing about how the timestamps are used to control the slave clock.
- The standard says nothing about how to correct the slave clock
  - It may immediately be overwritten with the computed time
    → the clock steps forward (result is a time gap) or
    → the clock steps backward (clock runs through a time period again)
  - Its oscillator may be accelerated or slowed down for a while in order to achieve a smooth correction
    → How fast should this happen?

Zurich University of Applied Sciences

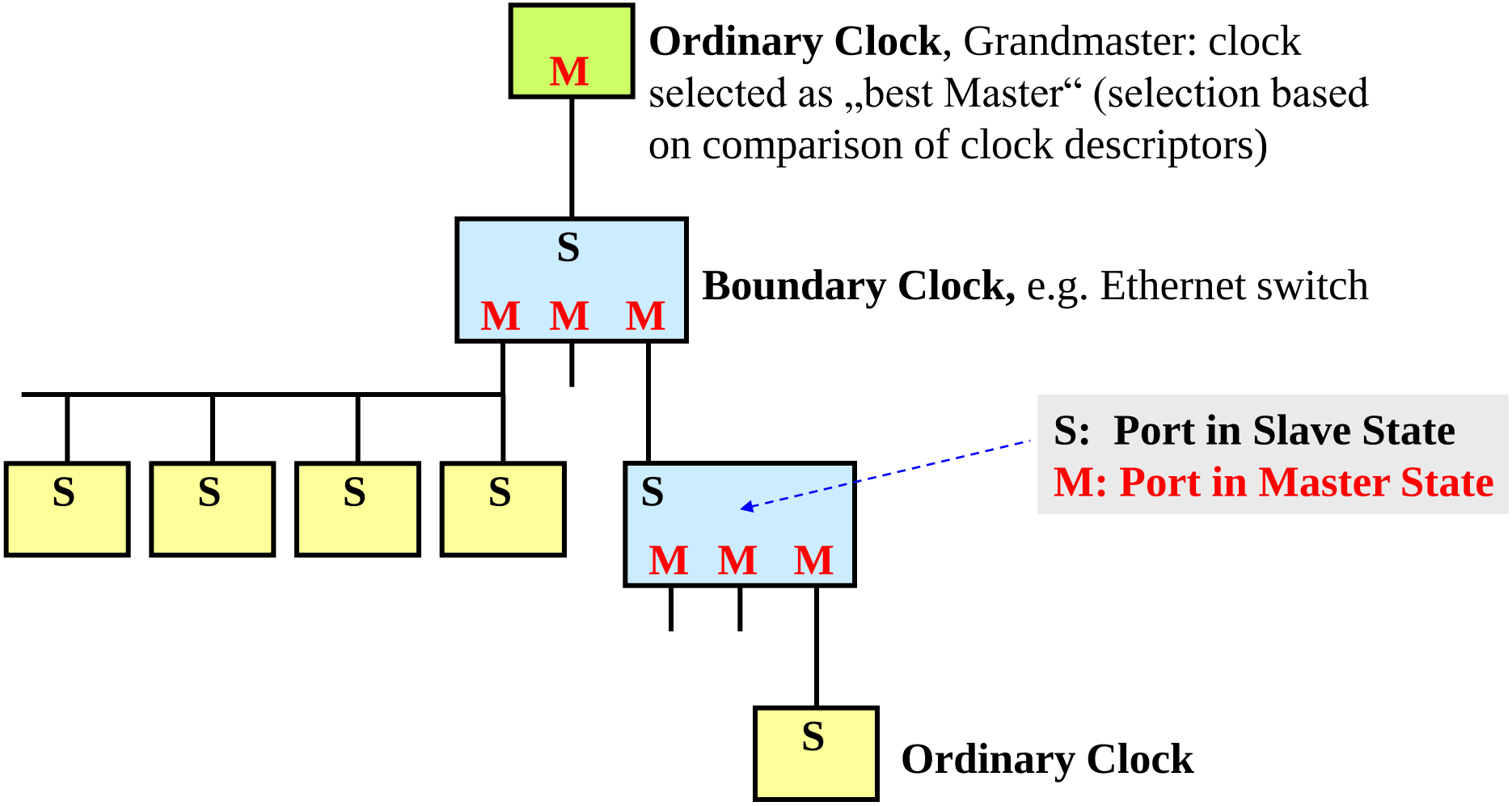
2 c) PTP-aware Infrastructure

Boundary Clock (BC)
PTP-aware Time Bridge



Boundary Clock (BC)

Best Master Clock Algorithm establishes a Synchronization Tree



Boundary Clock (BC)
Best Master Clock Algorithm (BMCA)

Zurich University of Applied Sciences

- Clocks organize themselves – the BMCA establishes a master-slave hierarchy
- Potential grandmasters exchange Announce messages
- BMCA is executed on each OC/BC port by comparing data sets (i.e. is my own data set compared with the one received by Announce → which one describes the better clock?)
- The comparison is based on the following attributes in the respective order
  - Priority1: a configurable clock priority
  - clockClass: a clock 's traceability
  - clockAccuracy: a clock's accuracy
  - offsetScaledLogVariance: a clock's stability
  - Priority2: a configurable second order clock priority
  - clockIdentity: a clock's unique identifier (the tie-breaker if all other attributes are equal)
- Based on this comparison a state decision is taken → the port state is set to either MASTER, SLAVE, or PASSIVE

Transparent Clock (TC)
Motivation

- **Deep cascaded topologies (daisy chained Boundary Clocks) represent a number of consecutive control loops → such a configuration is susceptible to jitter accumulation**

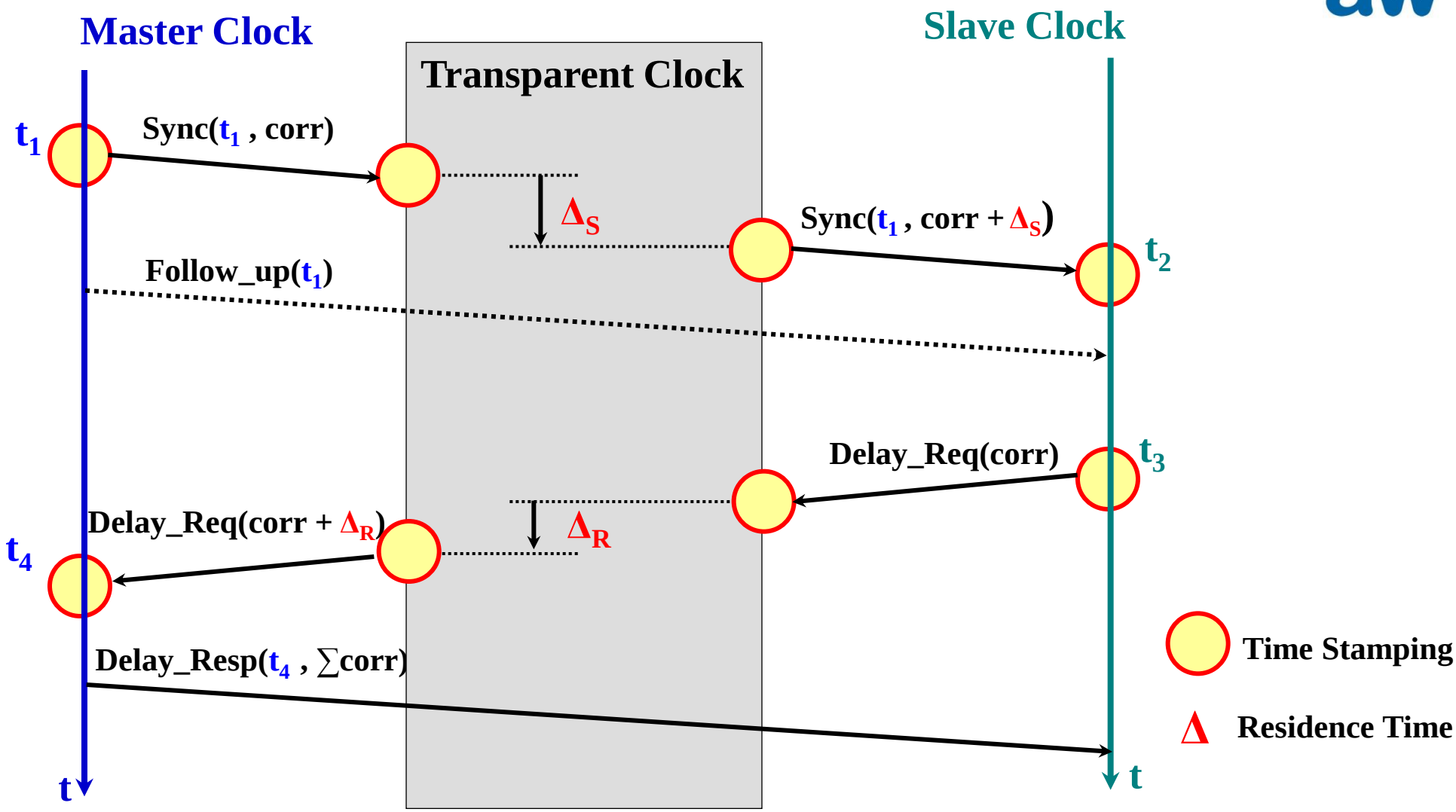
Chain is a popular topology in some applications.

Transparent Clock (TC)
Basic Idea

Zurich University of Applied Sciences

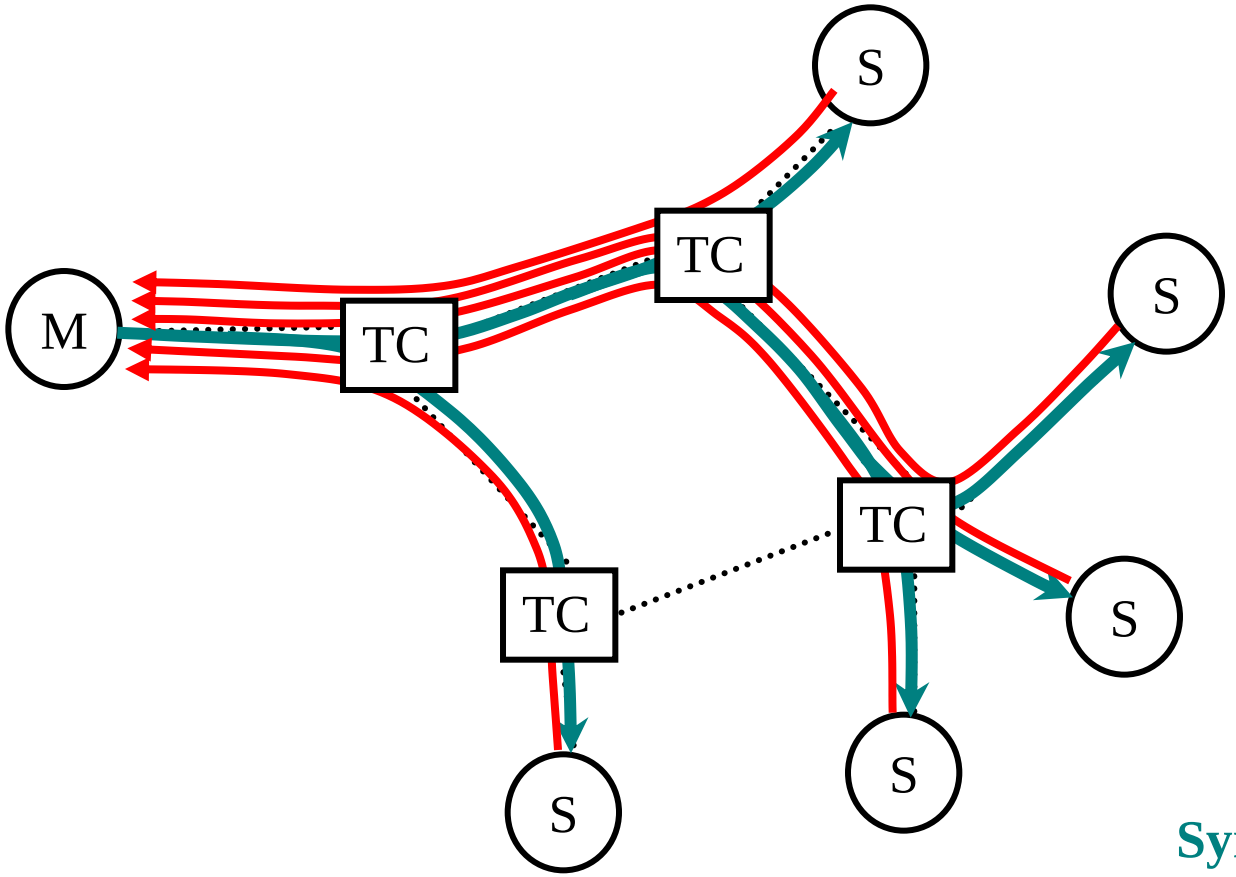
- **The concept of TC was introduced with PTP version 2**
- **The basic idea of the TC is to measure the time a PTP event message has spent in the switch (the so-called residence time)**
- **The residence time is reported to the receiver by the message itself or by a subsequent Follow_up or Delay_Resp message**
- **In IEEE 1588 version 2 a new message field has been added for this purpose, the so called Correction Field**
  - **it is of type TimeInterval (high resolution fixpoint value, least significant bit weights 2^{-16} ns, i.e. about 15 fs)**
  - **it can be used to accumulate residence time along the path of the message**
  - **it may also be used for other kind of corrections (e.g. asymmetry compensation)**

Transparent Clock (TC)
Principle Operation



Transparent Clock (TC)

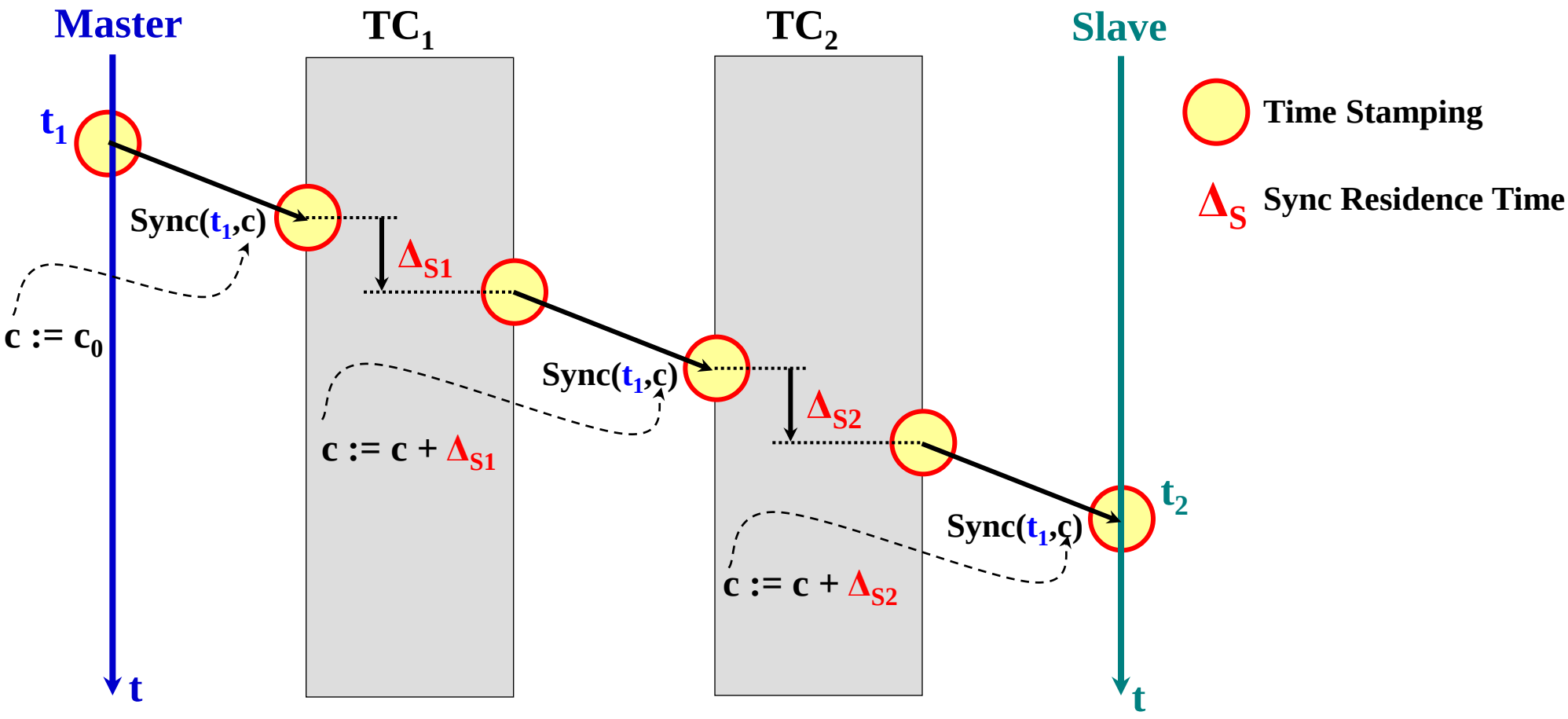
End-to-End Delay Measurement



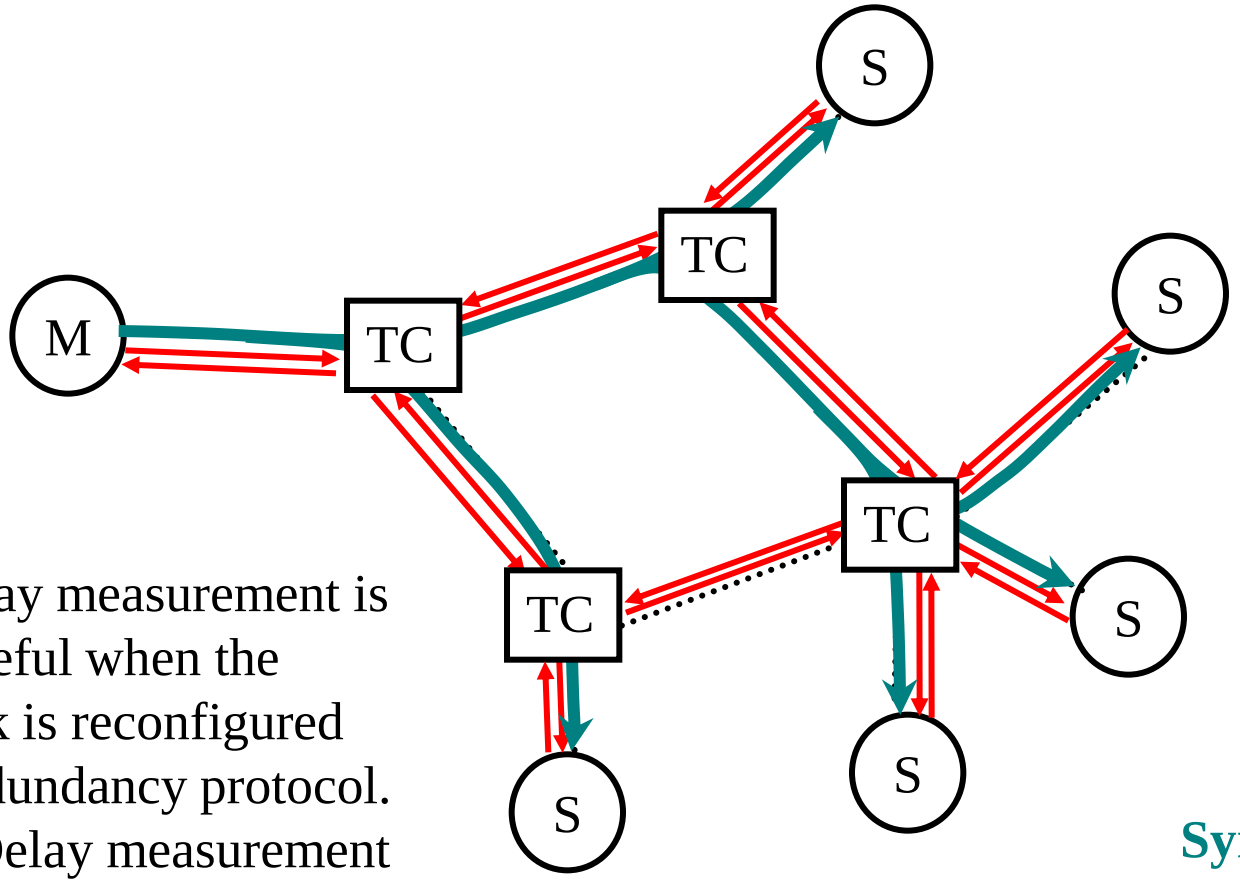
Sync Stream

e2e Delay Measurement

Transparent Clock (TC)
End-to-End Delay Measurement



Transparent Clock (TC)
Peer-to-Peer Delay Measurement

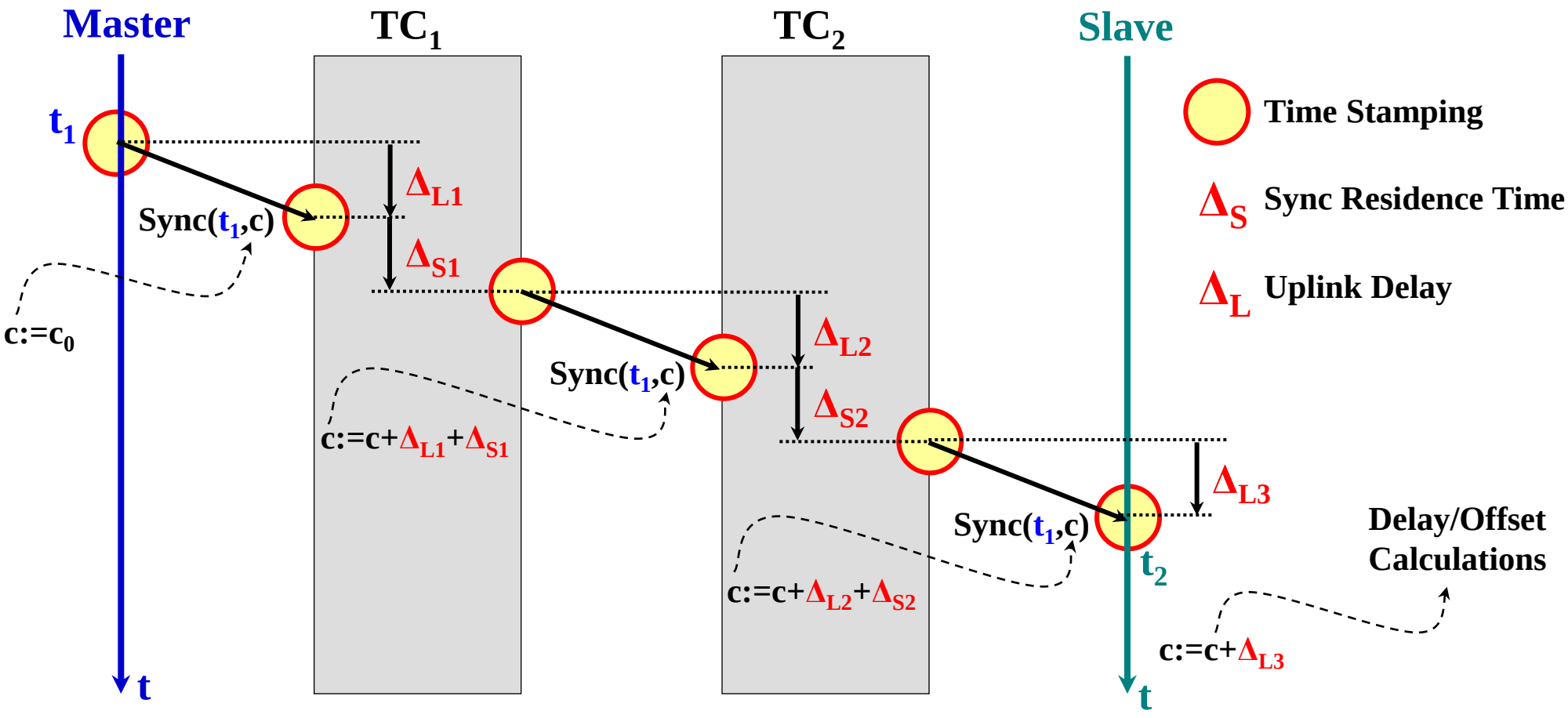


p2p delay measurement is very useful when the network is reconfigured by a redundancy protocol.
Note: Delay measurement over blocked ports!

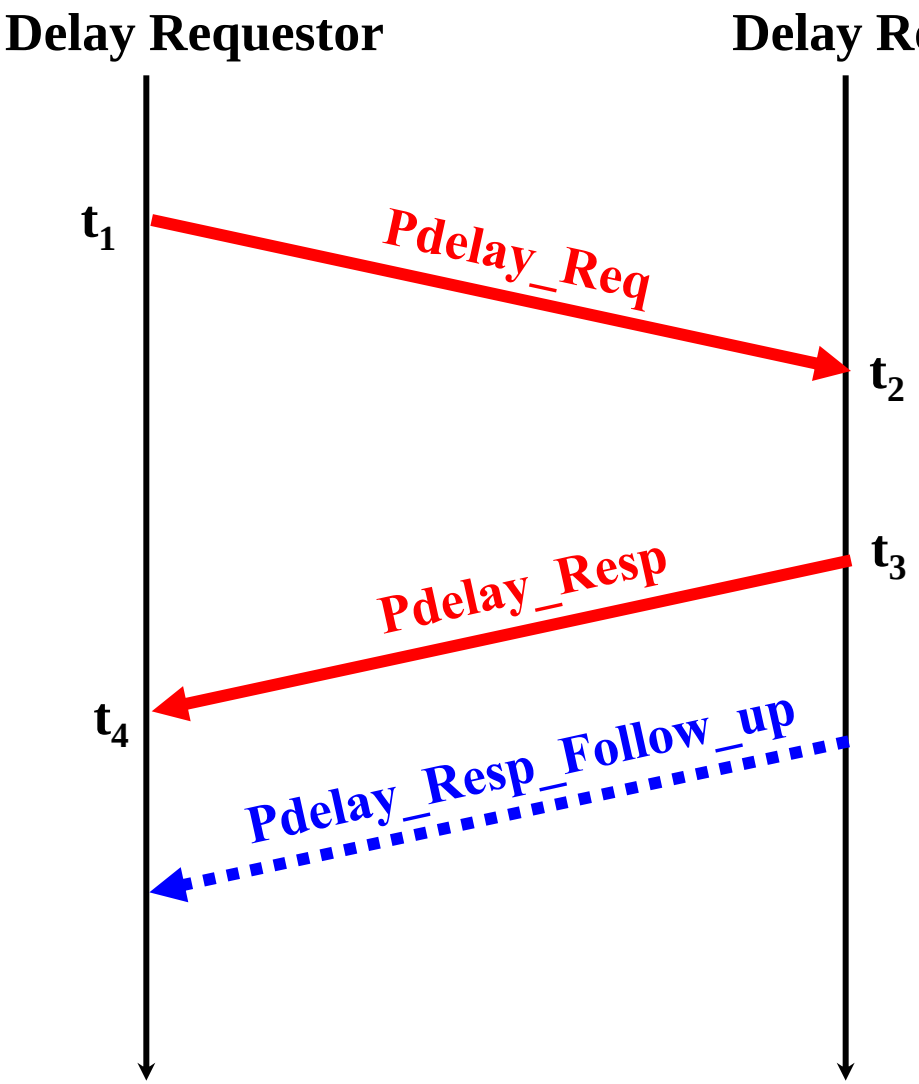
Sync Stream
p2p Delay Measurement

Transparent Clock (TC)

Peer-to-Peer Delay Measurement



Transparent Clock (TC)
Peer-to-peer Delay Measurement



Path Delay =$[(t_2-t_1)+(t_4-t_3)]/2$

Both peers initiate the link delay measurement independently

Three options for conveying timestamps to the requestor

a) **Pdelay_Resp(t_3-t_2)**
b) **Pdelay_Resp_Follow_up (t_3-t_2)**
c) **Pdelay_Resp(t_2)**
   Pdelay_Resp_Follow_up (t_3)

Infrastructure Components

PTP Device Type Comparison

Zurich University of Applied Sciences

zh aw

	BC	e2e TC	p2p TC	Legacy
Operational Principle	Hierarchy of clocks	Sum of switch residence time along the path is reported to the slave. Delay measurement end-to-end between slave and master.	All link delays are measured on a peer-to-peer basis. Sum of switch residence time **and link delay** along the path is reported to the slave.	Network is not PTP-aware. PTP functions in end nodes only
Variants *)	1-step / 2-step (per port)	1-step / 2-step	1-step / 2-step	n/a
Scalability	High (except in large shared media networks)	Poor (high load for master)	High if one-step Poor if two-step and multicast	Poor for multicast Unicast as alternative
Suitability	Star and tree topologies	Small configurations	Deep cascaded topologies	Restricted

IEEE 1588 Network
PTP Device Type Summary

Zurich University of Applied Sciences

- There are five types of PTP devices, as follows:
  - Ordinary Clock
  - Boundary Clock
  - End-to-end Transparent Clock
  - Peer-to-peer Transparent Clock
  - Management node
- Ordinary and Boundary Clocks are characterized by the following attributes:
  - priority1
  - priority2
  - clockClass
  - clockAccuracy
  - timeSource
  - offsetScaledLogVariance
  - numberPorts

3) IEEE 1588 version 2

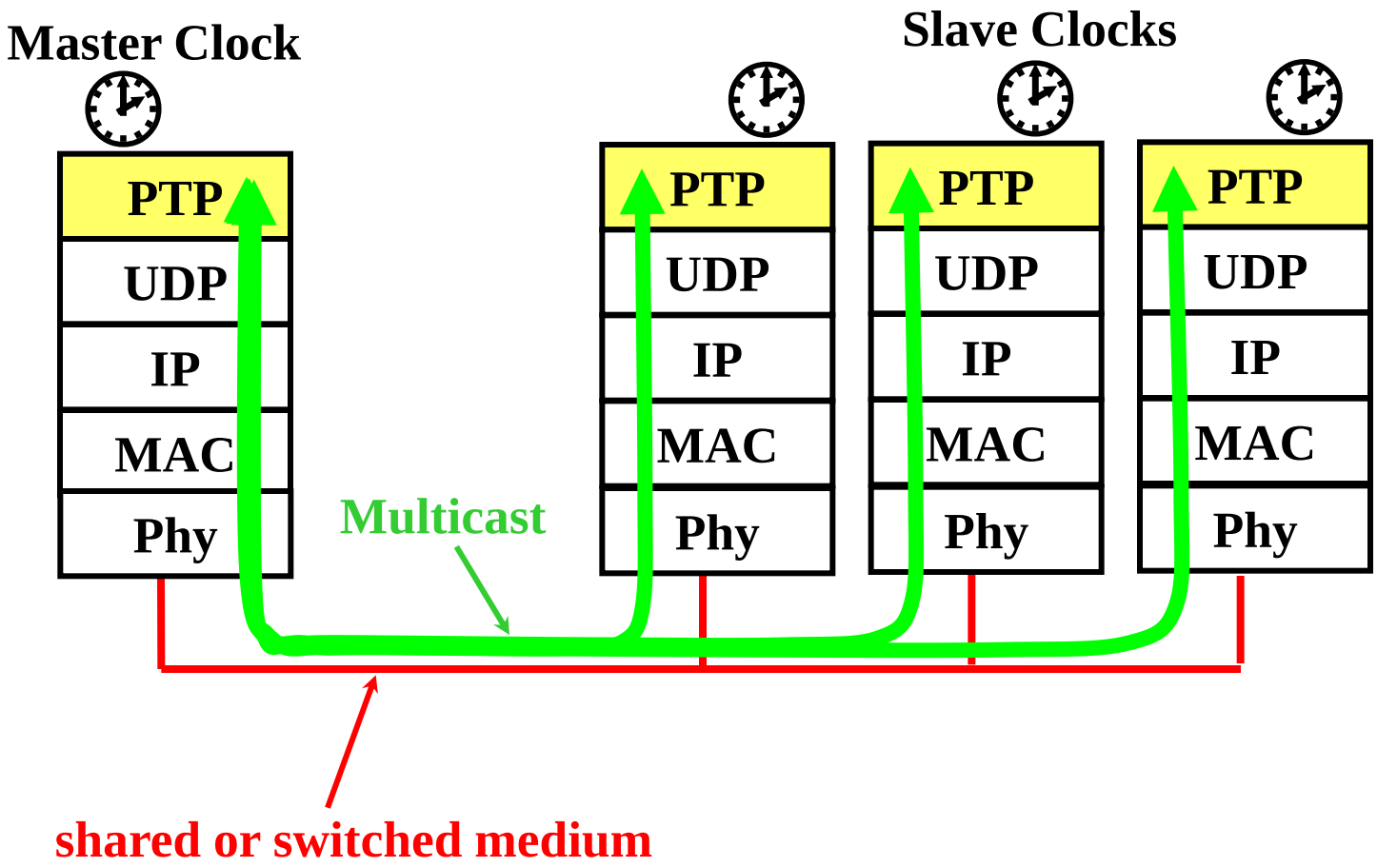
3 a) Message Flow, Mappings, and Addressing

No.	Time	Source	Destination	Protocol	Info
31	12.451282	192.168.100.181	224.0.1.129	PTPv2	Follow_Up Message
32	12.451459	192.168.100.181	224.0.1.129	PTPv2	Announce Message
33	13.551318	192.168.100.181	224.0.1.129	PTPv2	Sync Message
34	13.551653	192.168.100.181	224.0.1.129	PTPv2	Follow_Up Message
35	14.711437	192.168.100.181	224.0.1.129	PTPv2	Sync Message
36	14.711770	192.168.100.181	224.0.1.129	PTPv2	Follow_Up Message
37	14.711953	192.168.100.181	224.0.1.129	PTPv2	Announce Message
38	15.327622	192.168.100.203	224.0.1.129	PTPv2	Delay_Req Message
39	15.329288	192.168.100.181	224.0.1.129	PTPv2	Delay_Resp Message
40	15.811129	192.168.100.181	224.0.1.129	PTPv2	Sync Message
41	15.811444	192.168.100.181	224.0.1.129	PTPv2	Follow_Up Message
42	16.911413	192.168.100.181	224.0.1.129	PTPv2	Sync Message

```
⊞ Frame 35 (86 bytes on wire, 86 bytes captured)
⊞ Ethernet II, Src: DavicomS_7c:1f:e6 (00:60:6e:7c:1f:e6), Dst: IPv4mcast_00:01:81 (01:00:5e:00:01:81)
⊞ Internet Protocol, Src: 192.168.100.181 (192.168.100.181), Dst: 224.0.1.129 (224.0.1.129)
⊞ User Datagram Protocol, Src Port: ptp-event (319), Dst Port: ptp-event (319)
⊟ Precision Time Protocol (IEEE1588)
  ⊞ 0000 .... = transportSpecific: 0x00
    .... 0000 = messageId: Sync Message (0x00)
    .... 0010 = versionPTP: 2
    messageLength: 44
    subdomainNumber: 0
  ⊞ flags: 0x0200
  ⊞ correction: 0.000000 nanoseconds
    ClockIdentity: 0x00606effff7c1fe6
    SourcePortID: 1
    sequenceId: 27088
    control: Sync Message (0)
    logMessagePeriod: 0
    originTimestamp (seconds): 1228743652
    originTimestamp (nanoseconds): 157446500
```

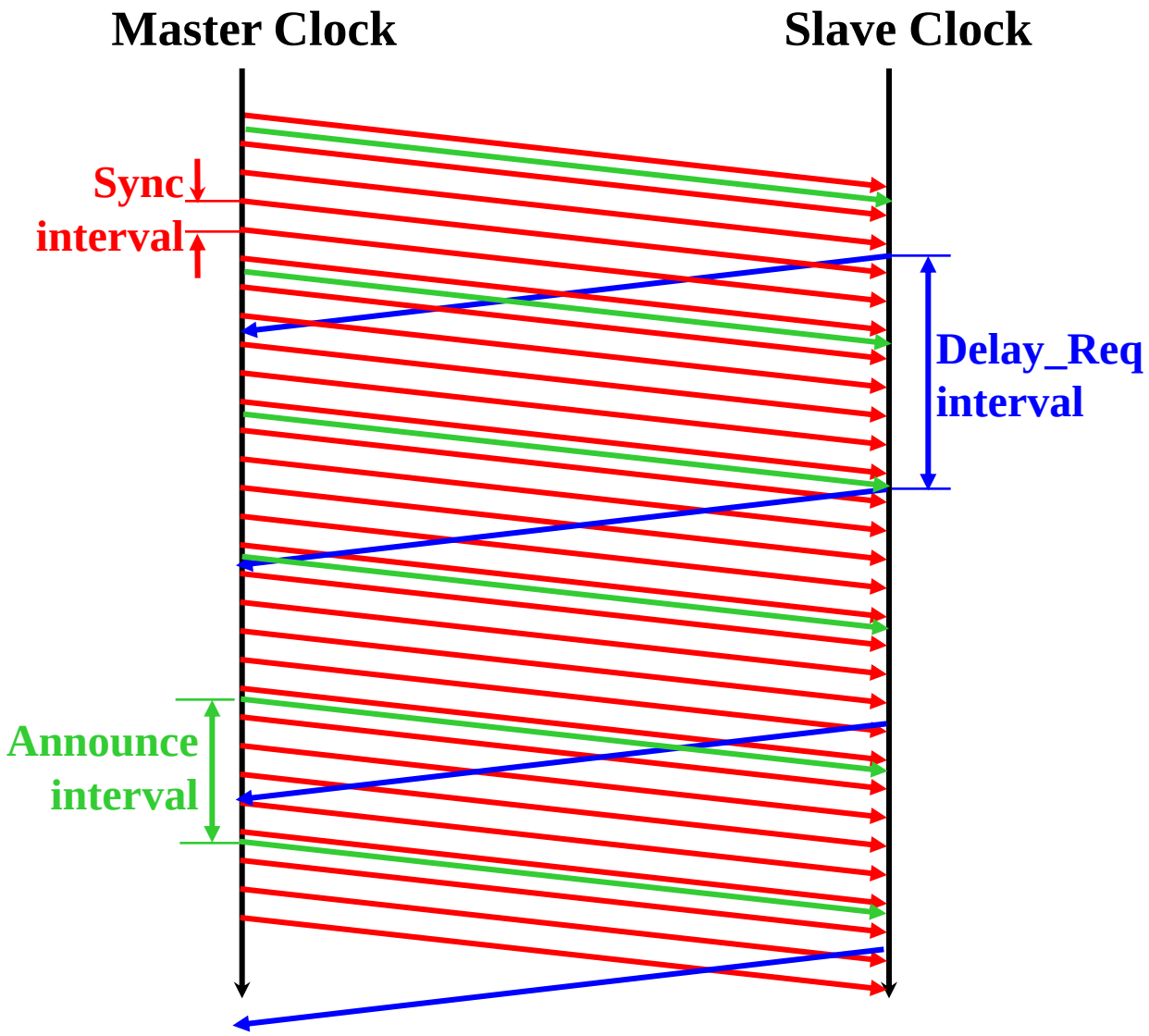
IEEE 1588 version 2

Per Default all PTP Communication is Multicast



IEEE 1588 version 2

Frequency of Message Exchange



Sync

- for Drift Compensation and Delay Measurement
- in a cyclic interval of 2^n s , n in the range of -7 to +6 , default is n=1 (2 s)
- optimal rate depends on oscillator stability

Announce

- carries the information needed to run the Best Master Clock Algorithm

Delay_Req

- for Delay Measurement
- in most cases much less frequent than Sync

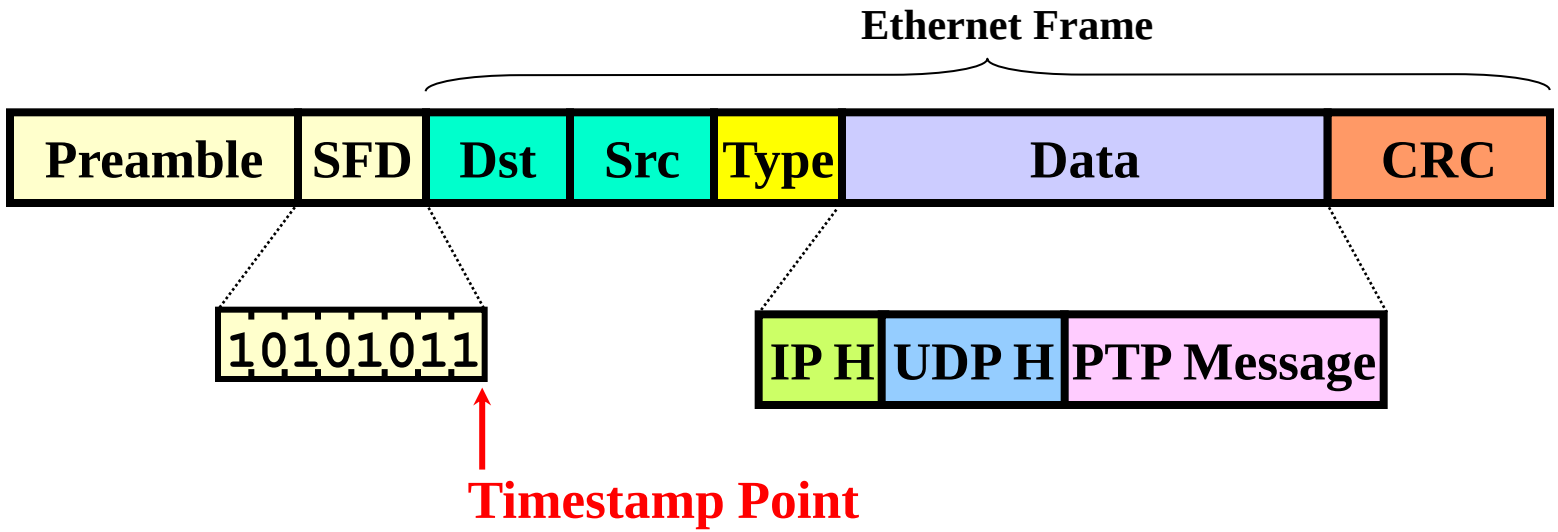
IEEE 1588 version 2
Sync Interval and Network Load

Zurich University of Applied Sciences

- The rate of each message type (i.e. Sync, Announce and Delay_Req) can independently be choosen
- The optimal frequency of clock adjustments depends on oscillator stability and expected accuracy
  - Sync and Delay_Req message rate can be up to 128 Sync messages per second
  - Be aware of the dimensions:
    - 1 ppm of oscillator deviation results in 1 µs per second (or 1s/12 days)
    - a cheap quartz has a temperature dependency of about 1 ppm/^{0}C or more
- Sync and Follow_up messages are sent as multicasts
  - the master can serve all slaves of a segment with one single message
  - each slave can calculate its drift and offset individually
- Delay_Req and Delay_Resp messages have point-to-point significance, but are sent per multicast as well (no address administration required)
  - because the delay is assumed not to change quickly, it is not measured as often as the offset, but the resulting network load grows with (number of slaves)^2

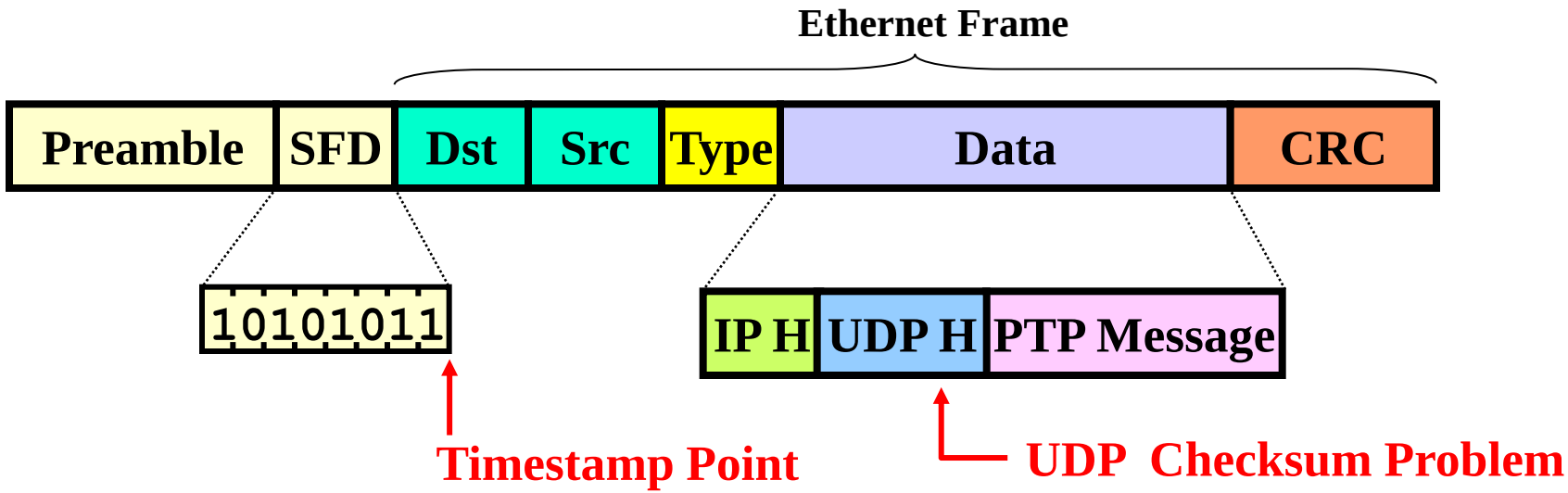
IEEE 1588 version 2
PTP over UDP/IPv4

- **UDP Port 319: event port for all messages that have to be timestamped**
  Port 320: general port for all other messages
- **IP multicast addresses 224.0.1.129 for all except peer delay measurement**
  224.0.0.107 for peer delay measurement
- **Ethernet Type field is 0x0800 (i.e. IP) and VLAN tag(s) may be present**



IEEE 1588 version 2
PTP over UDP/IPv6

- **UDP Port 319: event port for all messages that have to be timestamped**
  Port 320: general port for all other messages
- **IP multicast addr FF0x:0:0:0:0:0:0:181 for all except peer delay measurement**
  FF02:0:0:0:0:0:0:6B for peer delay measurement
  x is the address scope (e.g. x=4 means Admin-Local , x=2 means Link-Local)
- **Ethernet Type field is 0x0800 (i.e. IP) and VLAN tag(s) may be present**



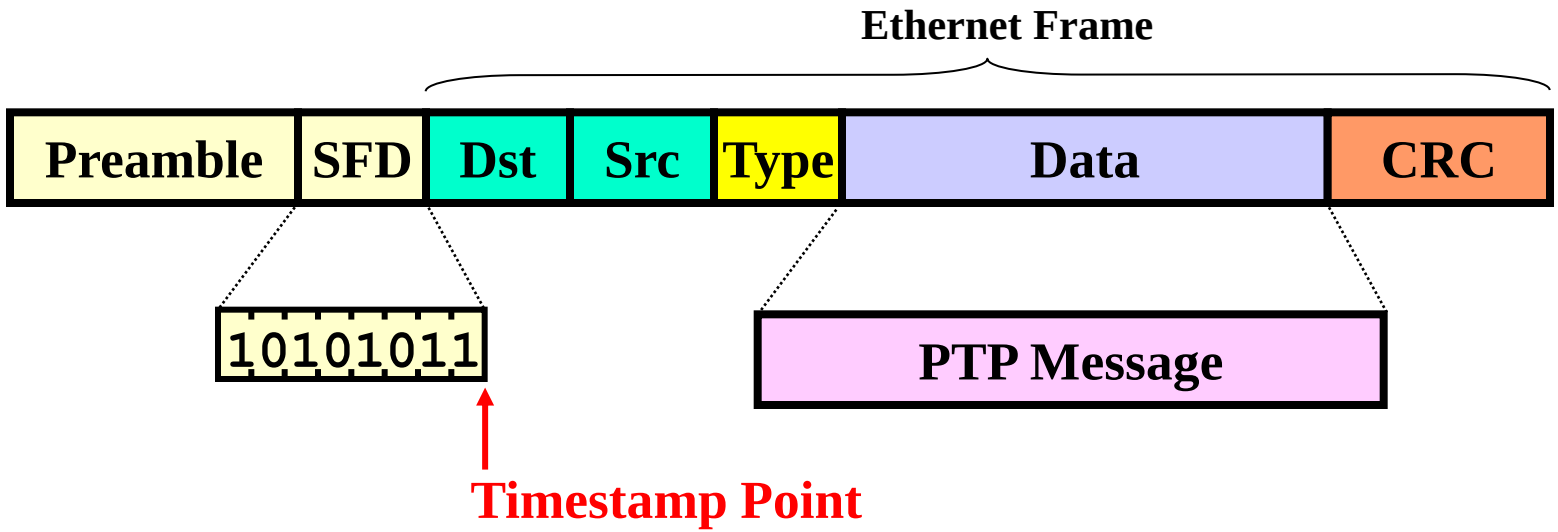
IEEE 1588 version 2

PTP directly over Ethernet (no UDP/IP)

Zurich University of Applied Sciences

- **MAC**
  multicast addresses 01-1B-19-00-00-00 for all except peer delay measurement
  01-80-C2-00-00-0E for peer delay measurement

- **Ethernet Type field is 0x88F7 (i.e. PTP) and VLAN tag(s) may be present**



IEEE 1588 version 2
Considerations of modifying PTP Messages on-the-Fly

Zurich University of Applied Sciences

- **Insertion of timestamp is not critical**
- **Adding a time interval to the correction field delays the TX path because the high order octett of the correction field goes over the line before the low order part**
- **Hop-by-hop hecksum (i.e. Ethernet FCS) has to be corrected in all cases**
- **The use of UDP checksum is optional in IPv4, but mandatory in IPv6**
  - **the UDP hecksum can be nulled when an IPv4 messages is modified**
  - **the handling in IPv6 is tricky because the UDP checksum leaves a node before the modification is executed → Workaround: send the UDP checksum unmodified but set an adjustment field at the end of the modified message in such a way that the checksum turns out to be correct**
- **Some on-the-Fly operations are considered as layer violations of the OSI model and still subject of discussions in IEEE 802**

IEEE 1588 version 2

Common part of PTP Message Header

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
transportSpecific				messageType				1	0
reserved				versionPTP				1	1
messageLength								2	2
domainNumber								1	4
reserved								1	5
flags								2	6
correctionField								8	8
reserved								4	16
sourcePortIdentity								10	20
sequenceId								2	30
controlField								1	32
logMessageInterval								1	33

Source: IEEE 1588-2008, Table 18

IEEE 1588 version 2

PTP Message Type Field

Zurich University of Applied Sciences

zh aw

Message type	Message class	Value(hex)
Sync	Event	0
Delay_Req	Event	1
Pdelay_Req	Event	2
Pdelay_Resp	Event	3
Reserved	—	4-7
Follow_Up	General	8
Delay_Resp	General	9
Pdelay_Resp_Follow_Up	General	A
Announce	General	B
Signaling	General	C
Management	General	D
Reserved	—	E-F

Source: IEEE 1588-2008, Table 19

IEEE 1588 version 2

PTP Message Flag Field Values (Octet 0)

Zurich University of Applied Sciences

Octet	Bit	Message types	Name	Description
0	0	Announce, Sync, Follow_Up, Delay_Resp	alternateMasterFlag	FALSE if the port of the originator is in the MASTER state. Conditions to set the flag to TRUE are specified in subclauses 17.3 and 17.4.
0	1	Sync, Pdelay_Resp	twoStepFlag	For a one-step clock, the value of twoStepFlag shall be FALSE. For a two-step clock, the value of twoStepFlag shall be TRUE.
0	2	ALL	unicastFlag	TRUE, if the transport layer protocol address to which this message was sent is a unicast address. FALSE, if the transport layer protocol address to which this message was sent is a multicast address.
0	5	ALL	PTP profile Specific 1	As defined by an alternate PTP profile; otherwise FALSE
0	6	ALL	PTP profile Specific 2	As defined by an alternate PTP profile; otherwise FALSE
0	7	ALL	Reserved	See Note

Source: IEEE 1588-2008, Table 20

IEEE 1588 version 2

PTP Message Flag Field Values (Octet 1)

Zurich University of Applied Sciences

Octet	Bit	Message types	Name	Description
1	0	Announce	LI_61	The value of leap61 of timePropertiesDS data set
1	1	Announce	LI_59	The value of leap59 of timePropertiesDS data set
1	2	Announce	currentUtcOffsetValid	The value of currentUtcOffsetValid of the timePropertiesDS data set.
1	3	Announce	ptpTimescale	The value of ptpTimescale of the timePropertiesDS data set.
1	4	Announce	timeTraceable	The value of timeTraceable of the timePropertiesDS data set.
1	5	Announce	frequencyTraceable	The value of frequencyTraceable of the timePropertiesDS data set.

Source: IEEE 1588-2008, Table 20

IEEE 1588 version 2
Management of PTP Clocks

Zurich University of Applied Sciences

- PTP is specified independent from the transport layers
- That's why a native management protocol is specified
- The management message structure is based on TLVs
- Management actions can be
  - GET or SET a data set or data set member
    → node answers with a RESPONSE
  - COMMAND to initiate an event
    → node answers with an ACKNOWLEDGE
- As an alternative to the native management protocol, an equivalent management mechanism may be specified (e.g. SNMP)
  - an SNMP MIB is defined for the power profile
  - the TicToc WG of the IETF has proposed a preliminary MIB definition for a PTP Slave-Only Ordinary clock compliant with the telecom profile G.8265.1
  - some switch vendors have implemented vendor specific MIBs

3 b) Timescales / Time Represetation

Source: leapsecond.com

Timescales / Time Represetation
Definition of a Second

- **Originally, the second was defined as 1/86'400 of a mean solar day as determined by the rotation of the Earth around its axis and around the Sun**
  - **The rotation is slowing by about 2 milliseconds per day because of tidal friction and other effects.**
- **In 1967 the International Committee of Weights and Measures adopted the following definition:**
  - **"The second is the duration of 9,192,631,770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the caesium-133 atom."**

Solar Clock

Chip Scale Atomic Clock

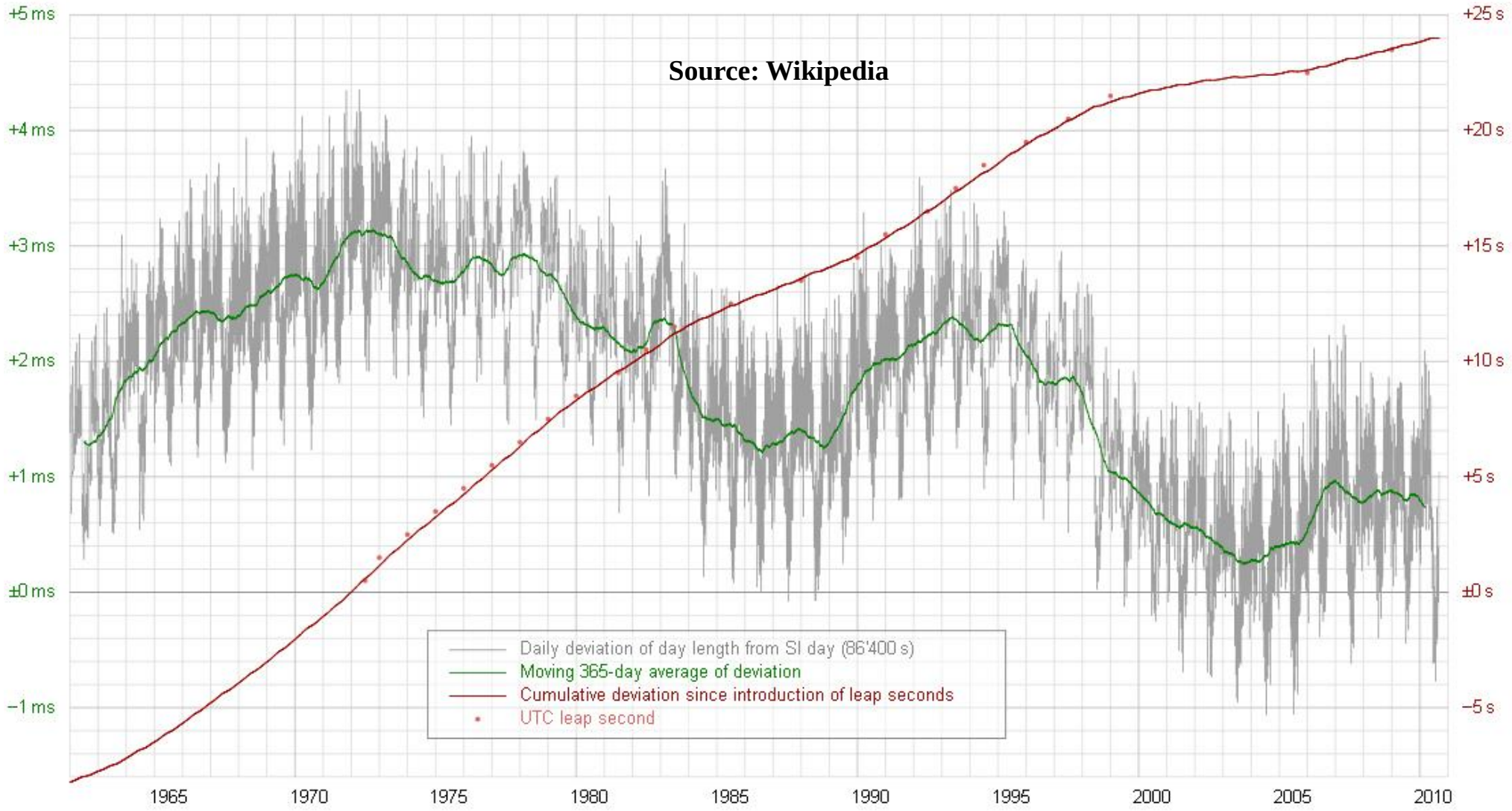
Timescales / Time Represetation

Duration of Earth Rotation

Zurich University of Applied Sciences

zh aw

Source: Wikipedia



Timescales / Time Represetation

TAI, UTC, and Leap Seconds

Zurich University of Applied Sciences

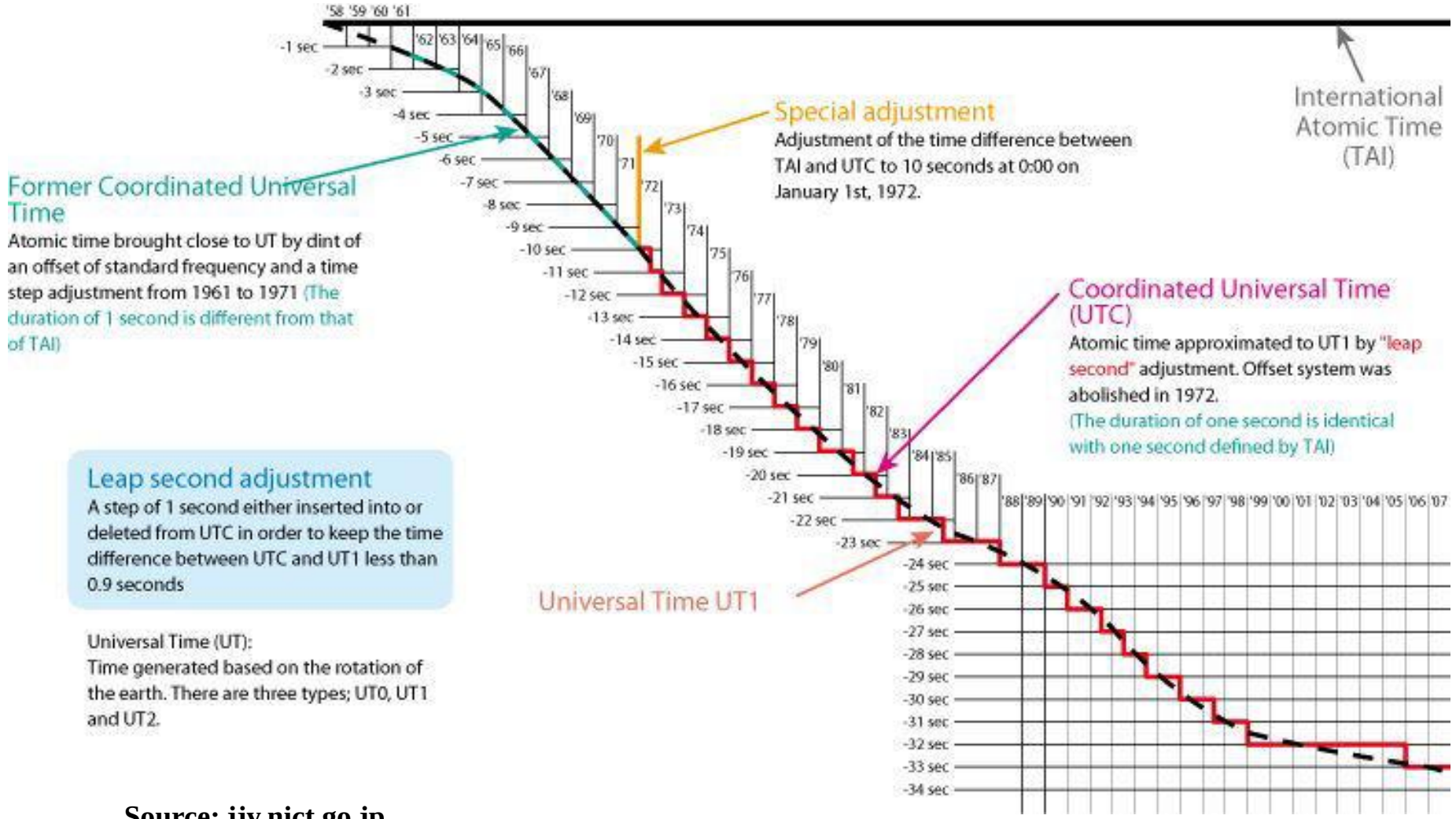
zh aw

- **TAI: Temps Atomic International (International Atomic Time)**
  - **advances continuously**
- **UTC: Universial Time Coordinated**
  - **experiences a discontinuities**
- **GPS time: delivered by GPS**

- **UTC, TAI, and GPS time advance identical (i.e. based on the same seconds definition)**
- **UTC time differs from GPS time and TAI by a constant offset. This offset is modified on occasion by adding or subtracting a leap second.**
- **PTP time is TAI**
  - **PTP distributes the current number of leap seconds in the currentUtcOffset field of the Announce message → UTC = TAI – currentUtcOffset**

Timescales / Time Represetation

TAI, UTC, and Leap Seconds



Source: jjy.nict.go.jp

Timescales / Time Represetation

PTP Time and Epoc

- The epoch is the origin of the timescale of a domain
- The PTP epoch coincides with the POSIX epoch and is defined as
  - January 1st 1970 00:00:00 TAI
- PTP time representation consists of a 48 bit seconds and a 32 bit nanoseconds field
- The PTP version 2 seconds field overflows after about 8‘925‘512 years
- PTP Announce delivers
  - currentUtcOffset in oder to allow conversion of PTP time to UTC
  - LI_59 and LI_61 in order to handle leap seconds
- Conversion between time systems
  - PTP_Seconds = NTP_Seconds - 2‘208‘988‘800 + currentUTCOffset
  - NTP_Seconds = PTP_Seconds + 2‘208‘988‘800 - currentUTCOffset
  - PTP_Seconds = GPS_Seconds + 315‘964‘819
  - GPS_Seconds = PTP_Seconds - 315‘964‘819

Time Representation in IEEE 1588 Version 2

- **Timestamp (data type for PTP clock and timestamps taken from it)**

```
struct Timestamp
{
  UInteger48 secondsField;
  UInteger32 nanosecondsField;
};
```

Note: The nanosecondsField member is always less than 10^9.

- **TimeInterval (data type for residence time and correction field)**

```
struct TimeInterval
{
  Integer64 scaledNanoseconds;
};
```

Note: The scaledNanoseconds member is the time interval expressed in units of nanoseconds and multiplied by 2^{+16}.

Timescales / Time Represetation

Time Domain

Zurich University of Applied Sciences

- **Within a domain, the characteristics of the time are determined by the grandmaster clock of the domain. These are:**
  - **The rate at which time advances**
  - **The origin of the timescale, the so called epoch**
- **Available timescales and epochs for use by the grandmaster clock are:**
  - **PTP timescale: Indicated by a timePropertiesDS.ptpTimescale value of TRUE. The epoch is the PTP epoch.**
  - **ARB timescale: Indicated by a timePropertiesDS.ptpTimescale value of FALSE. The epoch is specific to the implementation.**
- **The domain mechanism allows to maintain more than a single timescale within the same network cloud**
  - **PTP version 1 distinguishes the domains by different multicast addresses (4 addresses are reserved for this purpose)**
  - **PTP version 2 uses the same multicast address for all domains. An 8 bit domain number serves as distinction. The default domain number is 0.**
  - **The (unicast) telecom profile separates domains by unicast addressing**

Zurich University
of Applied Sciences

3 c) Profiles

IEEE 1588 version 2 Profiles

Definition for Profile

Zurich University of Applied Sciences

- A profile is a specific selection of attribute values and optional features of PTPv2 and should define
  - configuration and node management options
  - path delay measurement option
  - range and default values of all configurable attributes
  - required, permitted, or prohibited transport mechanisms
  - required, permitted, or prohibited options
  - required, permitted, or prohibited node types
  - procedures to verify conformance
- Profiles can be created by standards bodies, industry trade associations or other appropriate organizations
- Up to now, the following profiles have been specified:
  - two „Default PTP Profiles“ according to Annex J IEEE 1588-2008
  - the telecom profile for frequency transfer (ITU-T G.8265.1)
  - the profile for power systems applications (PSRC, IEEE PC37.238)
  - the LXI profile

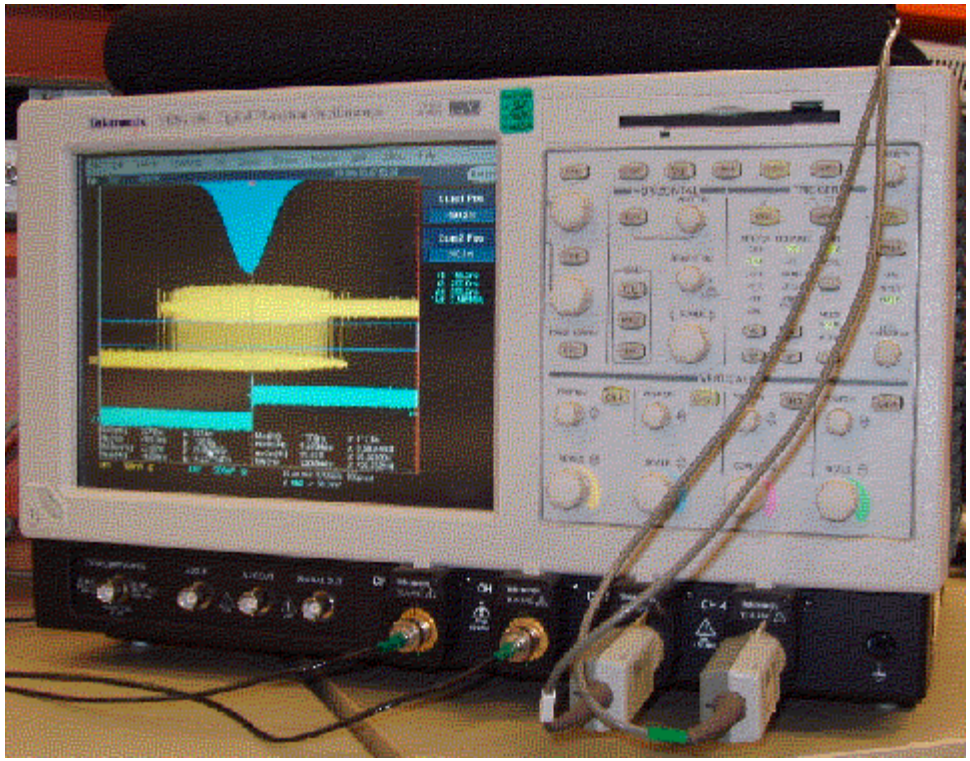
IEEE 1588 version 2 Profiles

Default Profiles

Zurich University of Applied Sciences

- **The Standard defines two Default Profiles**
  - **Delay Request-Response Default PTP profile (profile id: 00-1B-19-00-01-00)**
  - **Peer-to-Peer Default PTP profile (profile id: 00-1B-19-00-02-00)**
- **Both profiles specify**
  - **logAnnounceInterval is 0 to 4 (i.e. 1, 2, 4, 8, 16 s)**
  - **logSyncInterval is -1 to 1 (i.e. ½, 1, 2 s)**
  - **logMinDelayReqInterval / logMinPdelayReqInterval is 0 to 5 (i.e. 1, 2, 4, 8, 16, 32 s)**
  - **node management shall be implemented**
  - **BMC shall be implemented**
  - **permitted options are**
    - **the other delay measurement mechanism**
    - **unicast message negotiation**
    - **unicast discovery**
    - **grandmaster clusters**
    - **alternate master**
    - **acceptable master table**
    - **alternate timescales**
    - **path trace**

4) Implementing IEEE 1588



Implementing IEEE 1588

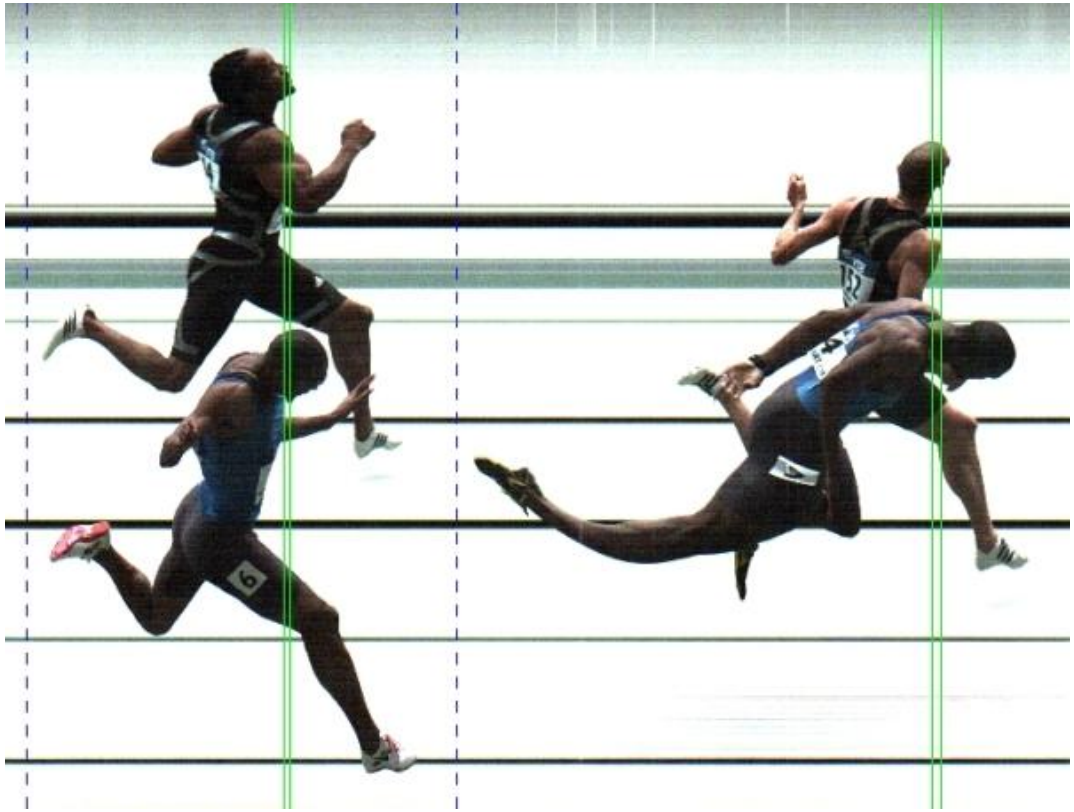
IEEE 1588 Hardware Assistance Implementation Options

Zurich University of Applied Sciences

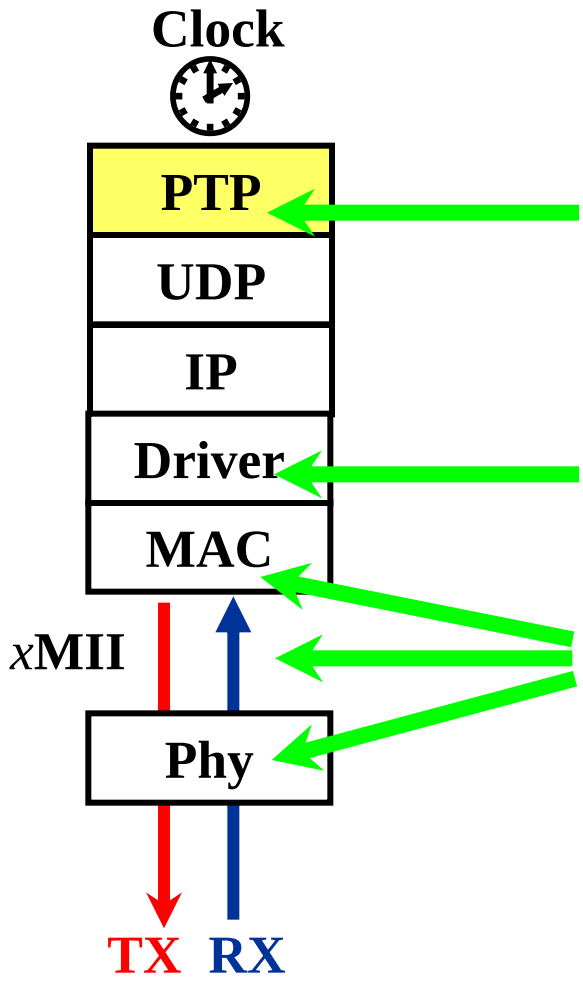
- FPGA tapping or intercepting the MII
- Ethernet PHY, e.g.
  - DP83640 100 Base-TX/-FX physical transceiver from National Semiconductor
  - 88E1340S quad 1000 Base-TX physical transceiver from Marvell
  - µPD60610 (µPD60620) single (dual) 10/100 Ethernet PHY from Renesas
- Ethernet Controller, e.g.
  - 82574, 82576, 82580 server class Ethernet controllers from Intel (GigE PHY, MAC, PCIe)
- Ethernet Switch, e.g.
  - LAN9313 3-Port Ethernet switch from SMSC
- Processor/Microcontroller, e.g.
  - many many processors and microcontrollers from Freescale
  - LM3S6950 microcontroller from Texas Instruments (former Luminary Micro)
  - BF518 (Blackfin) microcontroller/DSP from Analog Devices
- and an increasing number of other devices ☺

Zurich University of Applied Sciences

4a) Generation of Timestamps



Implementing IEEE 1588
Where to take Timestamps?

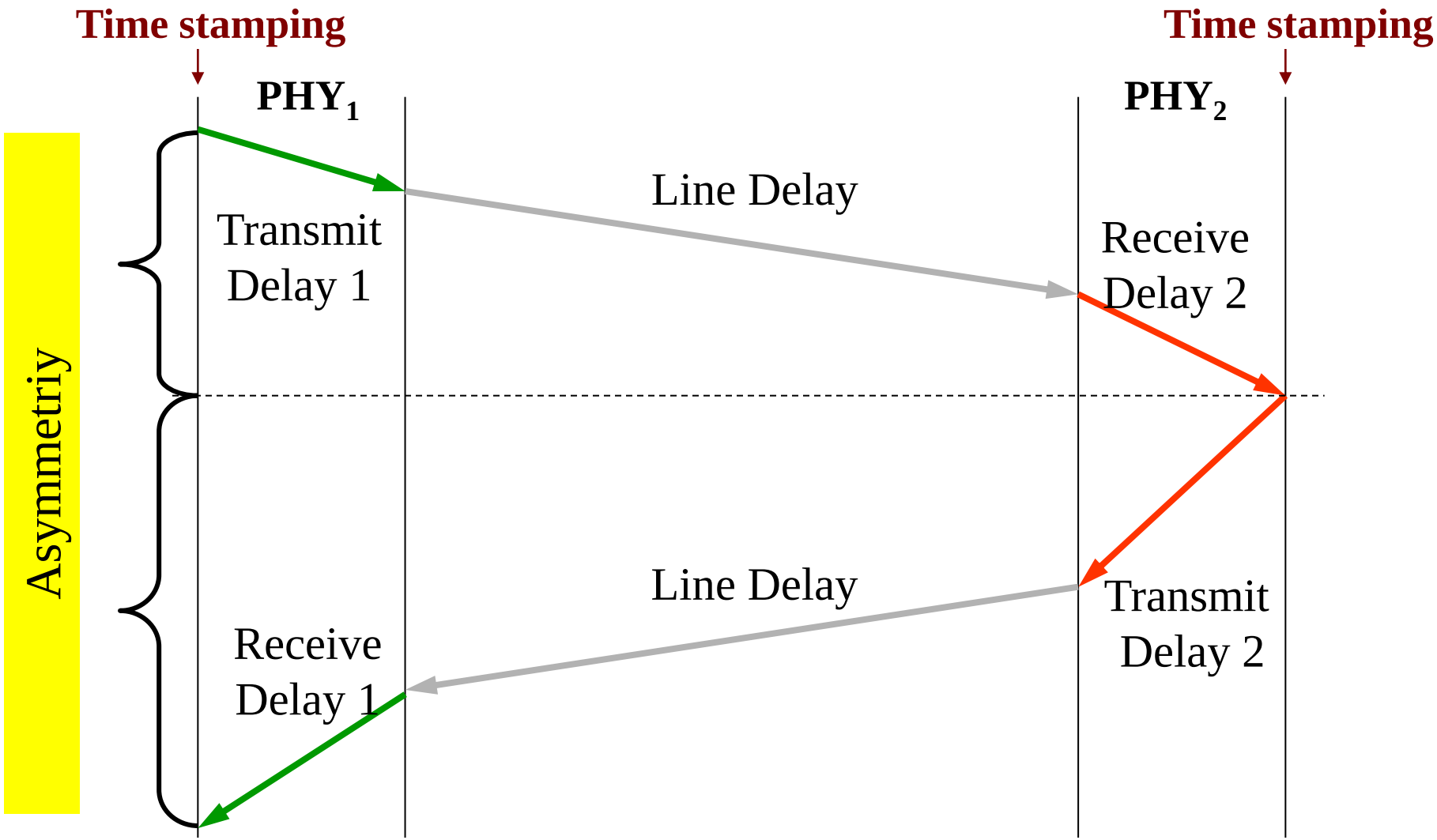


timestamping in the application layer (above socket)
→ exposed to delays in protocol stack and scheduling

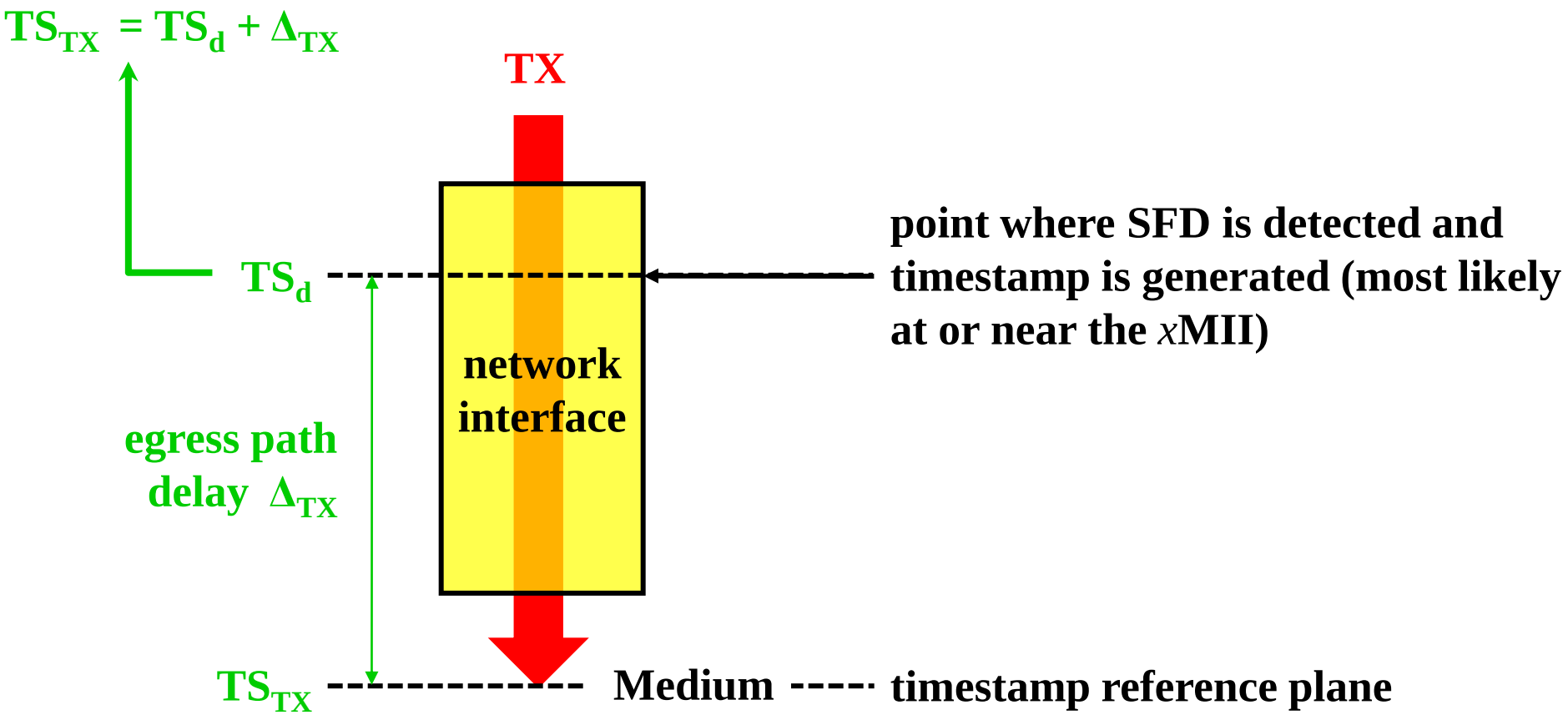
timestamping at network interrupt level
→ exposed to interrupt latency and network load

timestamping with hardware assistance
→most accurate timestamps
→tapping or intercepting the *x*MII or integrated into the PHY or the MAC function

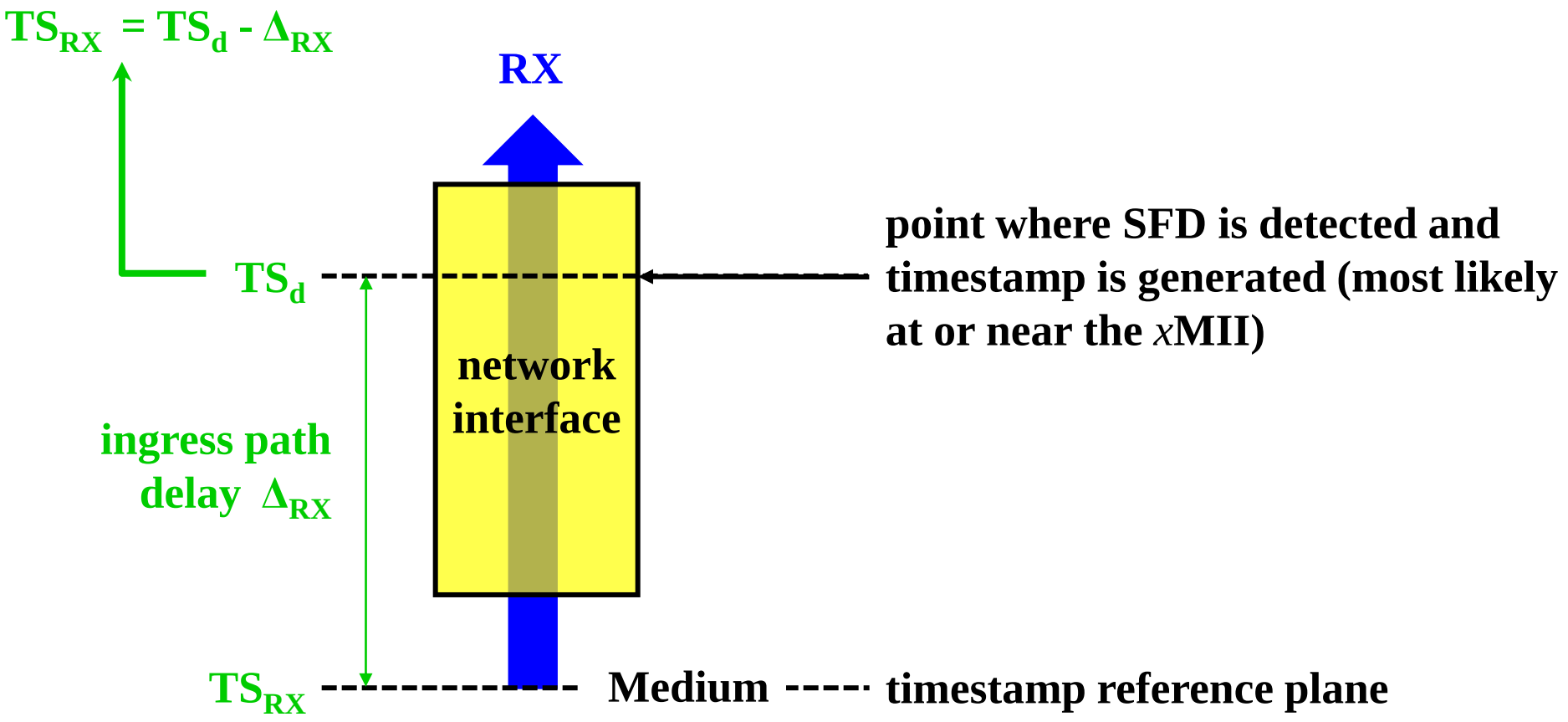
Implementation
PHY Path Delay Asymmetry



Implementing IEEE 1588
Egress Timestamping



Implementing IEEE 1588
Ingress Timestamping



Implementing IEEE 1588
Timestamp Precision

Zurich University of Applied Sciences

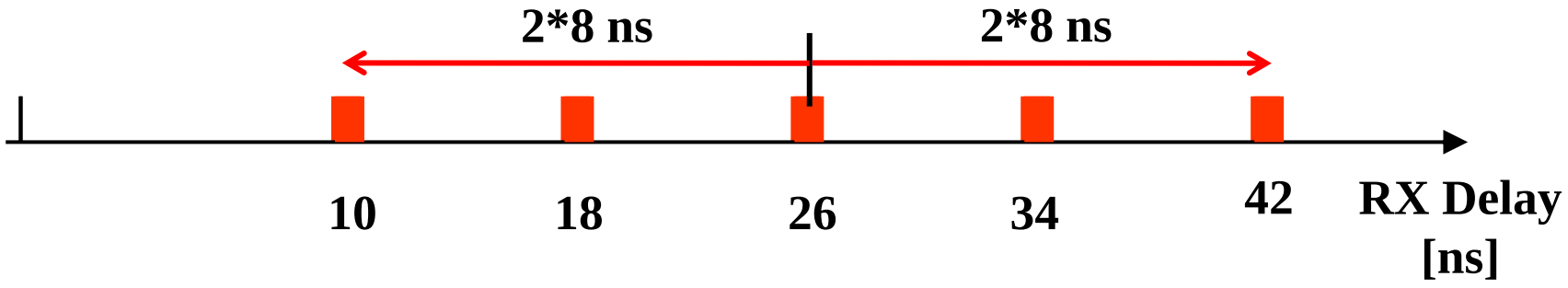
- **Timestamp precision is of paramount importance for syncronization accuracy**
- **One error source is timestamp quantization, i.e. the increment per clock cycle (e.g. +/- 4 ns @ 125 MHz clock)**
- **A PHY‘s path delay variation may be significantly higher than the clock jitter due to various processing steps at different clock rates between SFD detection and medium**
  - **FIFOs, PLLs**
  - **synchronization at clock domain crossings**
  - **coder/decoder**
  - **serializer/deserializer**

Implementing IEEE 1588
An Experience with Fast Ethernet PHYs

Zurich University of Applied Sciences

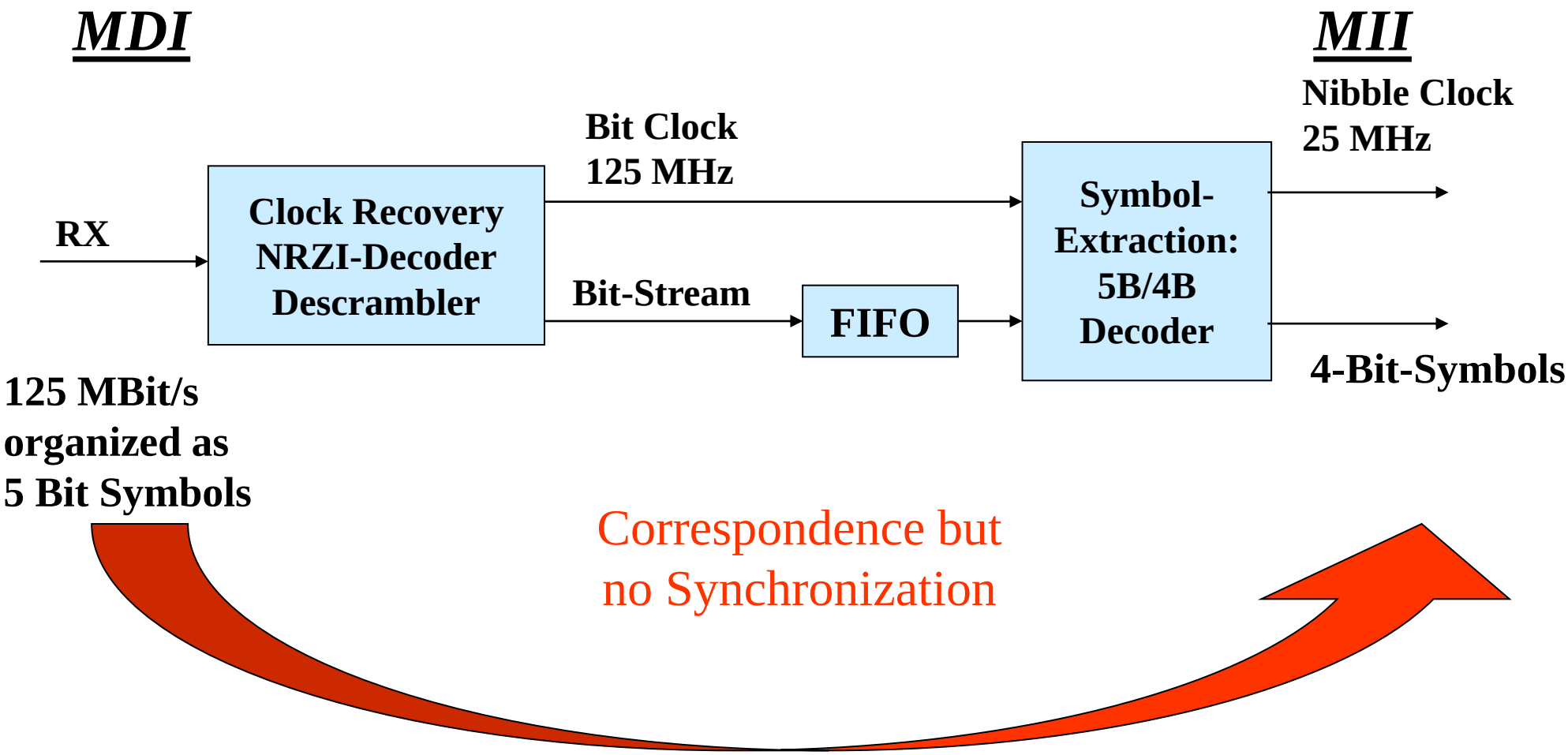
Example of a PTP-unfriendly PHY implementation

- An given master/slave pair showed stable synchronization but an unexpected offset from time to time
- It turned out that the RX path delay of the same PHY was not always the same
  - at link establishment the phase between the 25 MHz and the 125 MHz clocks locked in one of five possible positions in an 8 ns grid
  - this resulted in a timestamp uncertainty of ± 2*8 ns
  - the jitter during the lifetime of the link however was less than 1 ns



→ A PTP-friendly PHY has a well specified and constant RX and TY path delay

Implementing IEEE 1588
An Experience with Fast Ethernet PHYs



Implementing IEEE 1588
Characteristic of Timestamp Degradation

Zurich University of Applied Sciences

- **PHY path delay and path delay variation are implementation specific characteristics**
- **PHY path delay variation depends on how PLLs lock, when re-synchronization happens, how different clock domains work together. We can find PHY implementations where the path delay (latency between medium and timestamp point)**
  - **is constant over the lifetime of the component → PTP-frindley ☺**
  - **is constant over the lifetime of the link**
  - **may have changed after a period of inactivity (e.g. after wake-up of an Energy Efficient Ethernet link)**
  - **changes from frame to frame**
- **Delay and jitter generated by optical modules (i.e. SFP, XFP, SFP+, CFP) have to be considered as well**

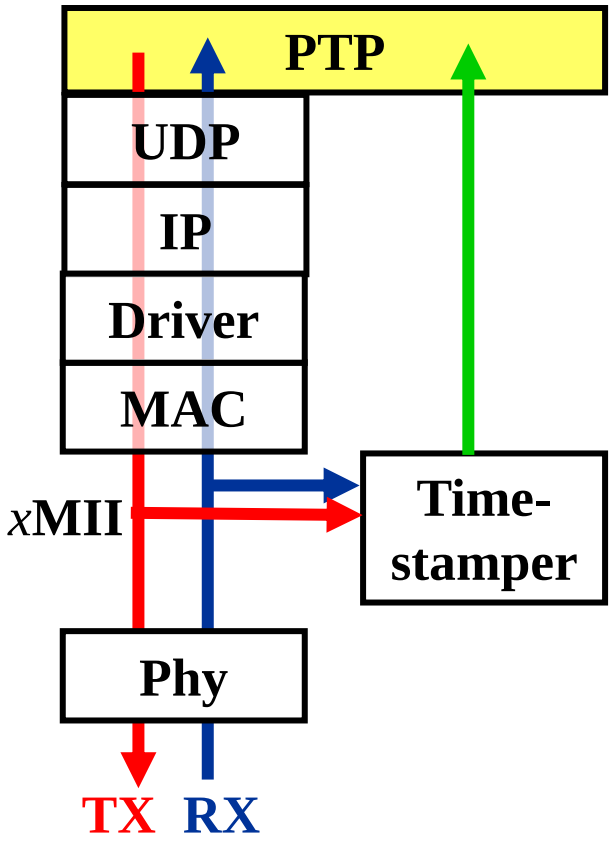
Implementing IEEE 1588

Delivering Timestamps to the Protocol Software

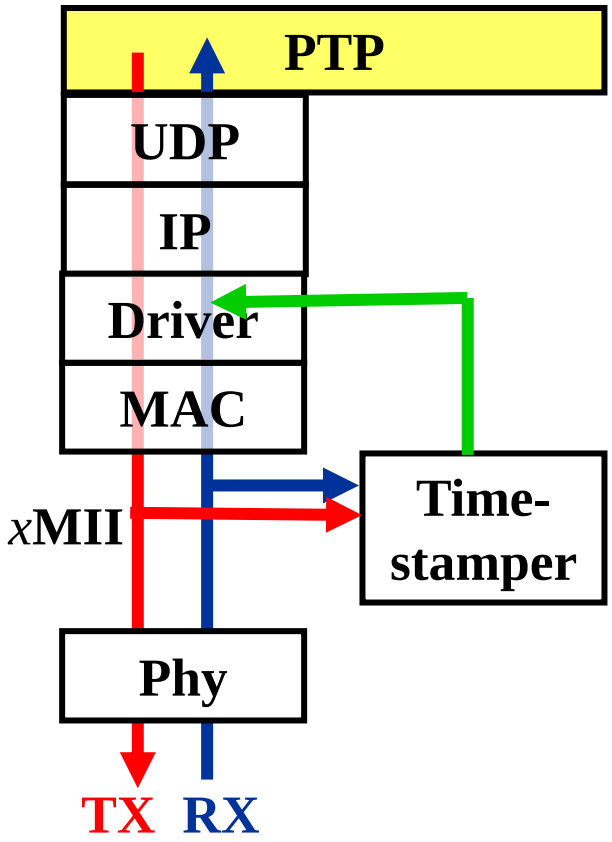
Zurich University of Applied Sciences

zhaw

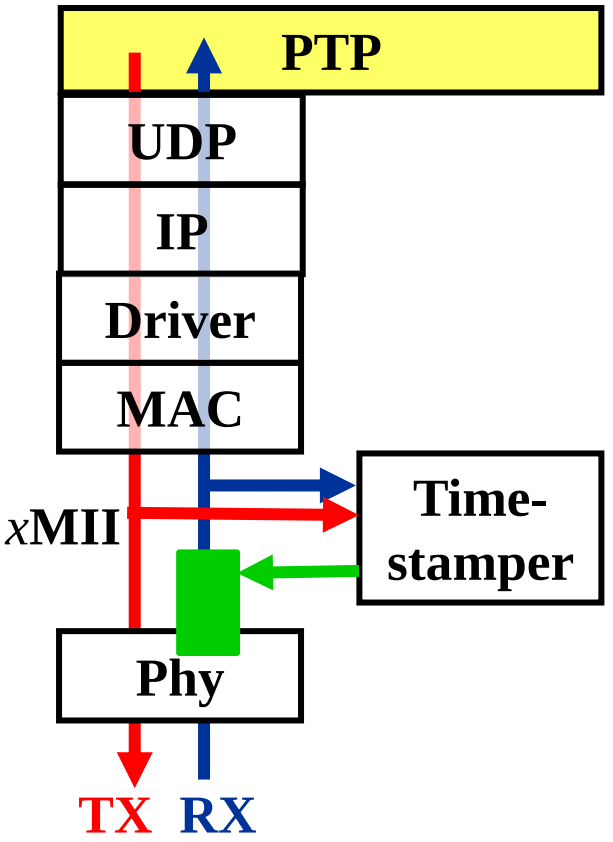
Option 1



Option 2



Option 3

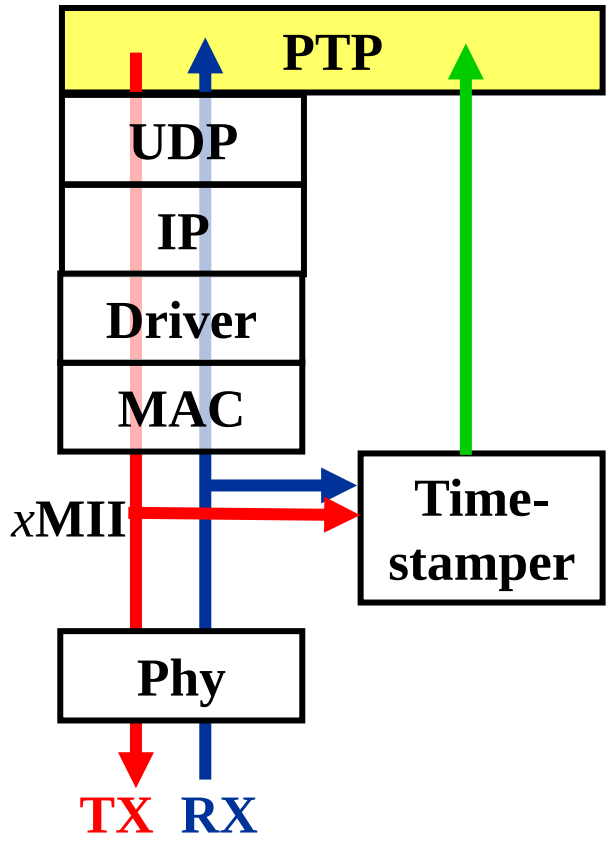


Implementing IEEE 1588
Timestamp Delivery – Option 1

Zurich University of Applied Sciences

Timestamp is delivered to the software independently of the frame

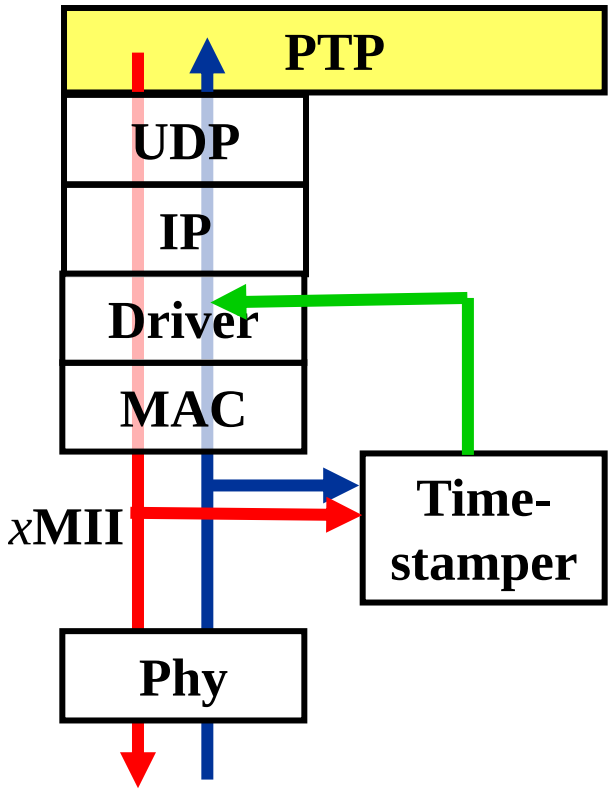
- **MAC and network driver do not need to be changed**
- **The timestamper has to generate a „fingerprint“ of the frame in order to allow the association of timestamp and corresponding frame**
- **The timestamper has to parse all frames in order to timestamp only frames of interest (e.g. PTP event messages). This parser may have to support multiple mappings.**
- **Unexpected timestamps may be generated (e.g. a received frame is timestamped but discarded later on due to some error condition)**



Implementing IEEE 1588
Timestamp Delivery – Option 2

Timestamp is delivered to the software together with the frame

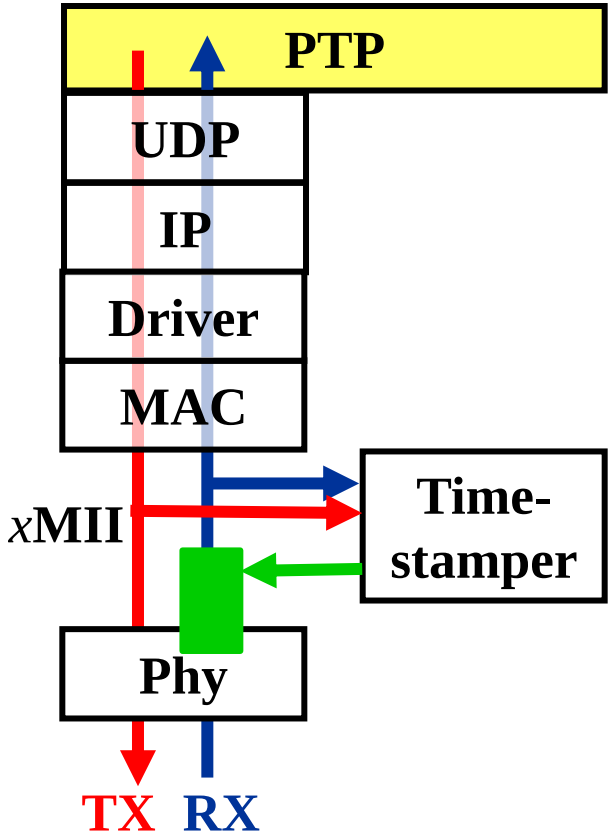
- **Timestamp is delivered in a field of the frame‘s datastructure (e.g. struct sk_buff in Linux)**
- **Application software can access timestamp using the socket option SO_TIMESTAMP or equivalent**
- **All received frames can be timestamped (timestamp generator does not need to parse the frames)**
- **Works for RX only, but TX timestamps can be delivered by looping back frames**
- **Linux follows this approach by defining a PTP-friendly timstamping interface (plus support of hardware clock)**



Implementing IEEE 1588
Timestamp Delivery – Option 3

Timestamp is delivered to the software in the frame itself

- **Some reserved bits in the PTP message are misused to insert the timestamp**
- **Only room for a part of the full 80 bit timestamp (most significant part is maintained in software)**
- **Message parser absolutely required**
- **Is only in RX direction applicable (may be sufficient for a slave only OC connected to a p2p TC)**
- **May be applied for OCs, but not for switches (frames should be forwarded unchanged)**



Implementing IEEE 1588
Timestamp Processing

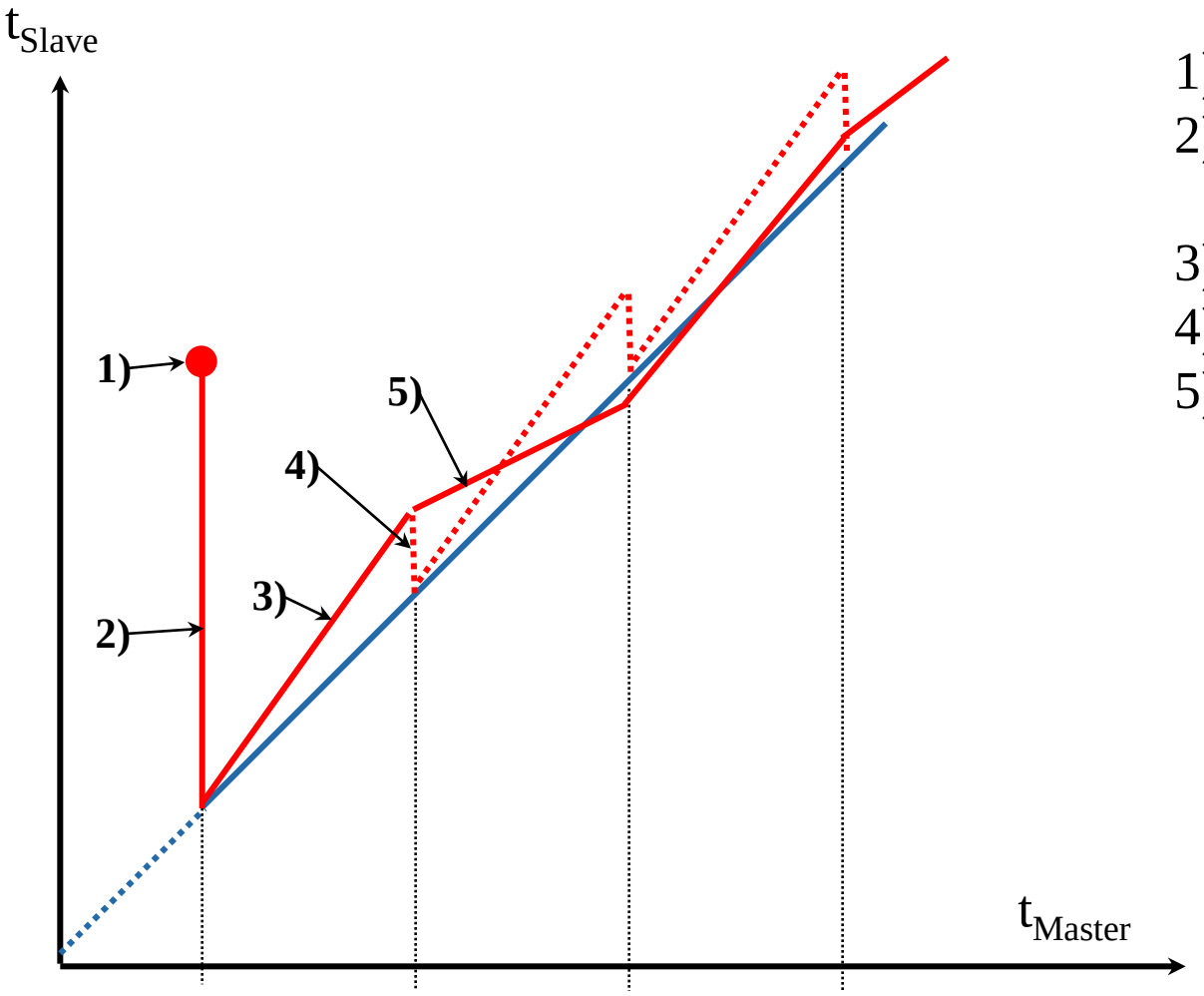
Zurich University of Applied Sciences

- **The raw timestamp needs to be corrected by the PHY path delay before further processing**
  - **this can be done in software in two-step operation**
  - **for one-step operation the correction has to happen on the fly – a register has to be provided for this purpose in hardware and needs to be set appropriately**

Zurich University
of Applied Sciences

4b) Maintaining PTP Time

Implementing IEEE 1588
Offset Correction Options



1) Slave starts with an arbitrary time
2) Slave‘s time is hardly set due to the large offset
3) The slave runs too fast
4) This results in a small offset
5) The slave slows down to compensate the difference by preserving monotonic growth of time

Remark 1: Jumping back has the disadvantage of running twice through the same interval

Remark 2: When not jumping back, it depends on the application how fast the offset should be compensated

Implementing IEEE 1588
Drift Compensation

Zurich University of Applied Sciences

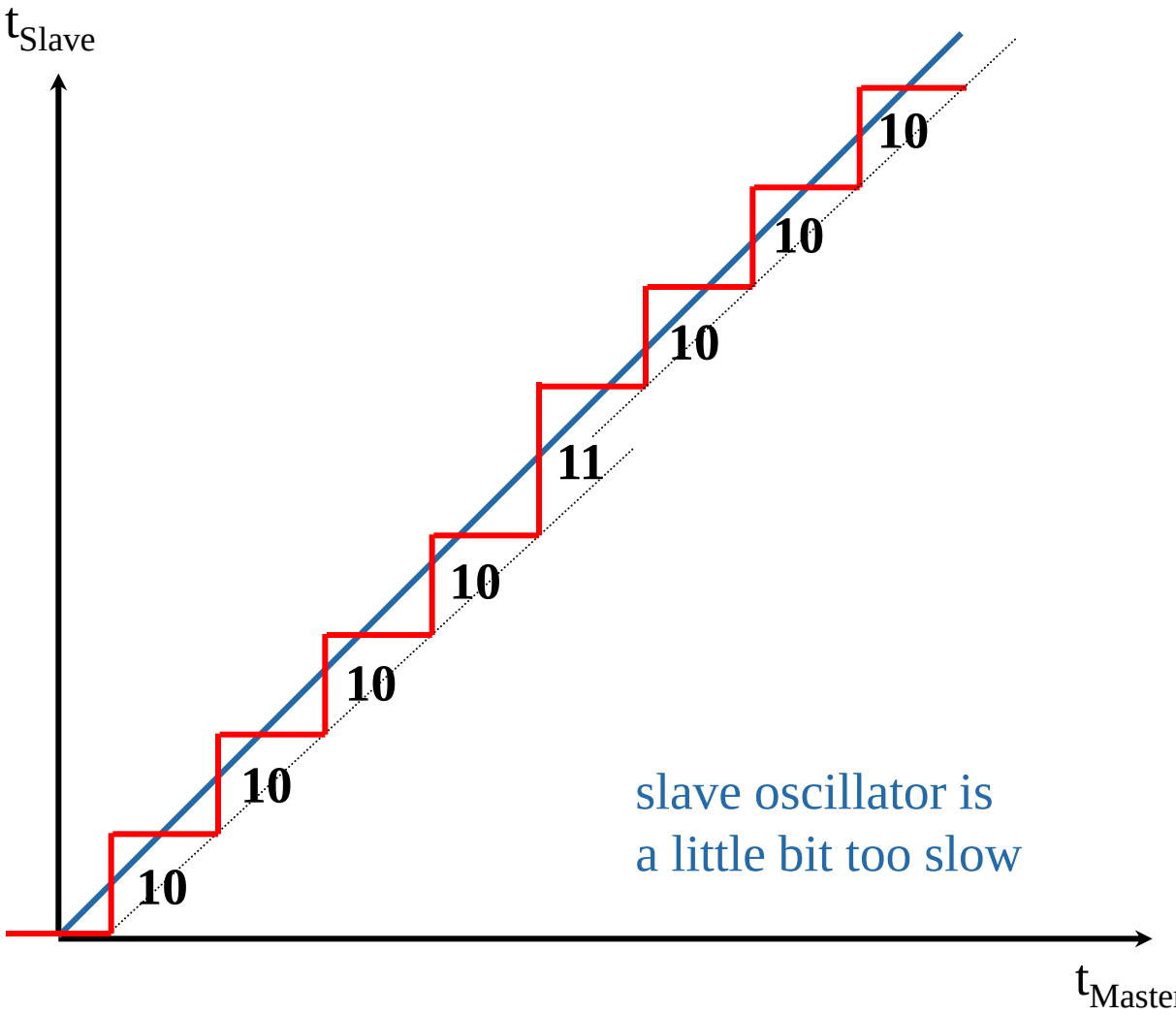
The default profile defines

- **Every grandmaster clock shall maintain a frequency deviating no more than 0.01% (100 ppm) from the SI second**
- **Any clock in the SLAVE state shall be able to correct its frequency to match any master clock meeting the above requirement**
  → The frequency adjustment range should be at least ±0.025% (±250 ppm)

There are several options to provide a tunable slave clock

1. **tunable oscillator: frequency is tuned by a control voltage (VCXO)**
   - **the DAC is subject to quantization**
   - **there is a tradeoff between DAC resolution, quantization error and pull-in range**
2. **Digitally tuned clock offers a simple and flexible solution**

Implementing IEEE 1588
Digital Drift Compensation

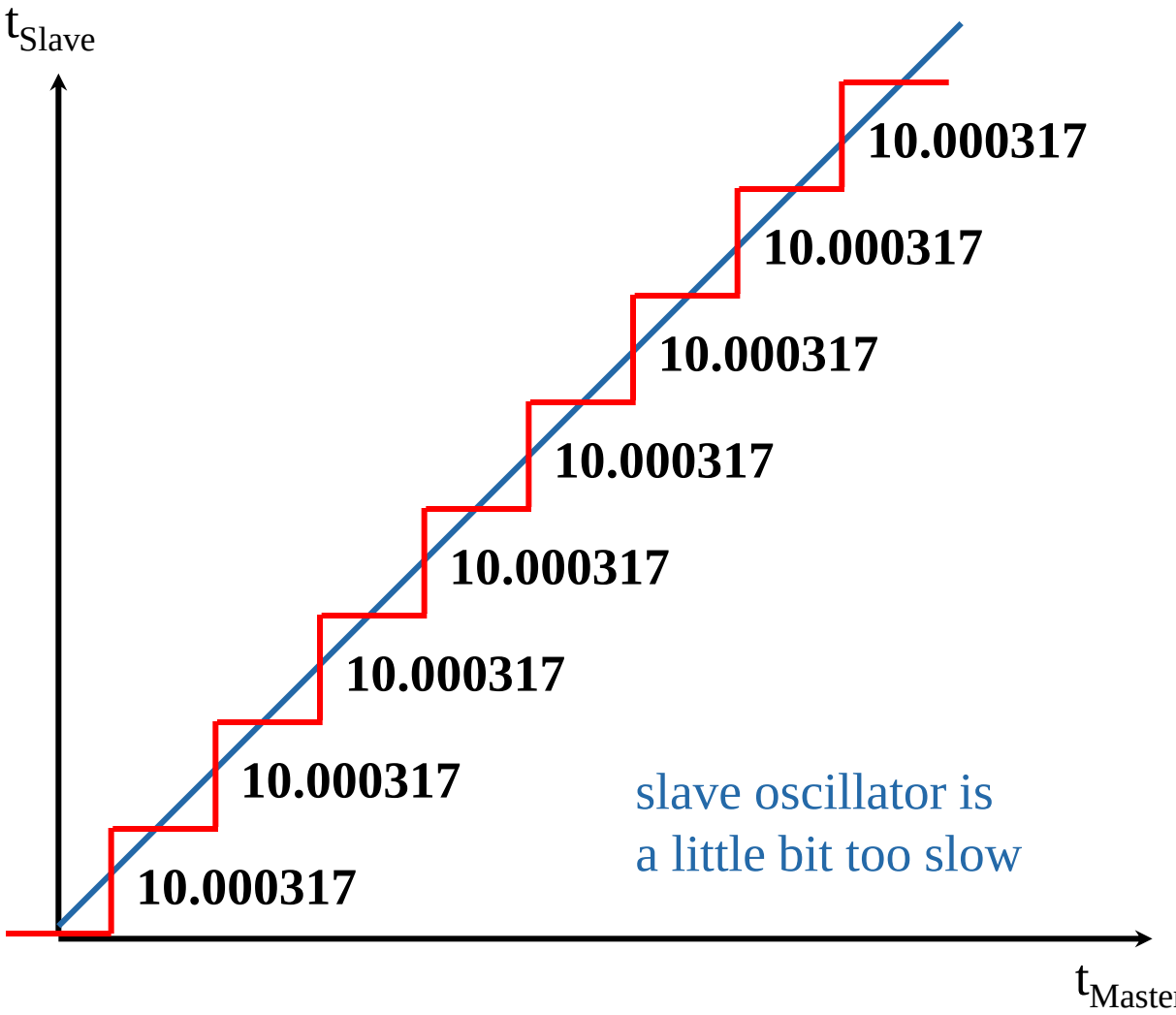


Example:
- 100 MHz free running oscillator
- nominal increment is 10 ns
- adjustment by adding
  9 or 11 ns every n^{th} cycle

Advantages
- easy to implement
- integer arithmetic

Disadvantages
- causes some jitter
- limited tuning capabilities
- not suitable for arbitrary oscillator frequencies

Implementing IEEE 1588
Digital Drift Compensation

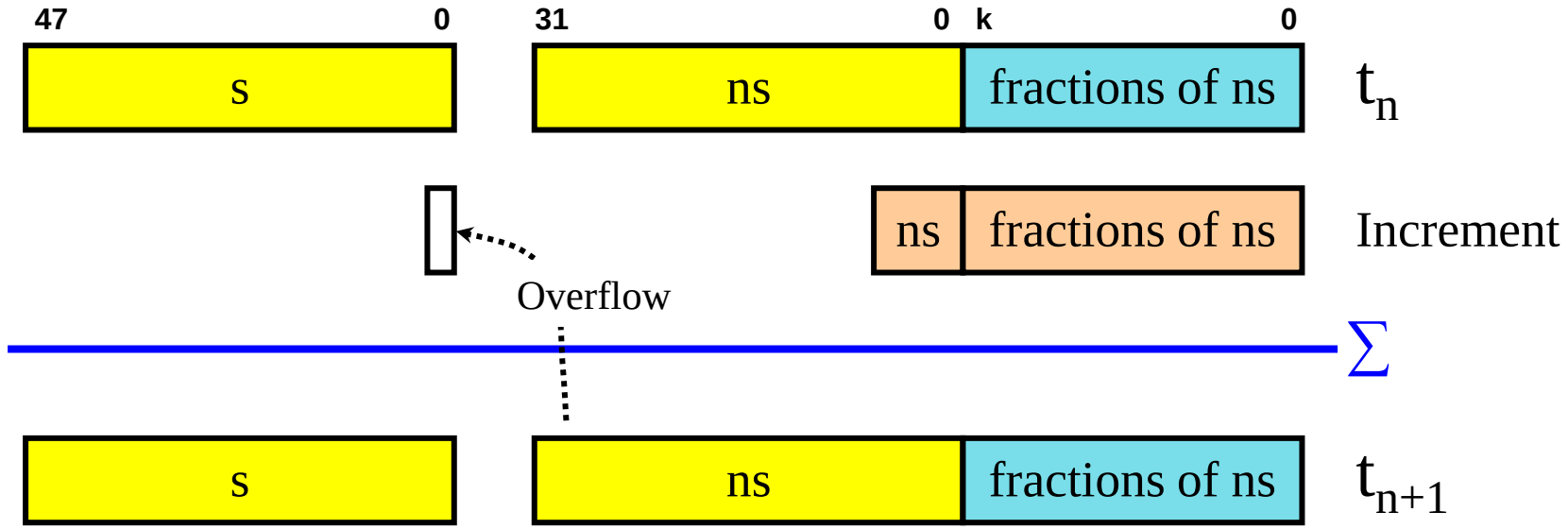


Example:
- 100 MHz free running oscillator
- nominal increment is 10 ns
- adjustment by adding the exact increment every cycle

Advantages
- smooth adjustment
- more flexible choice for oscillator frequencies

Disadvantages
- fixpoint arithmetic demands more resources

Implementing IEEE 1588
Digital Drift Compensation

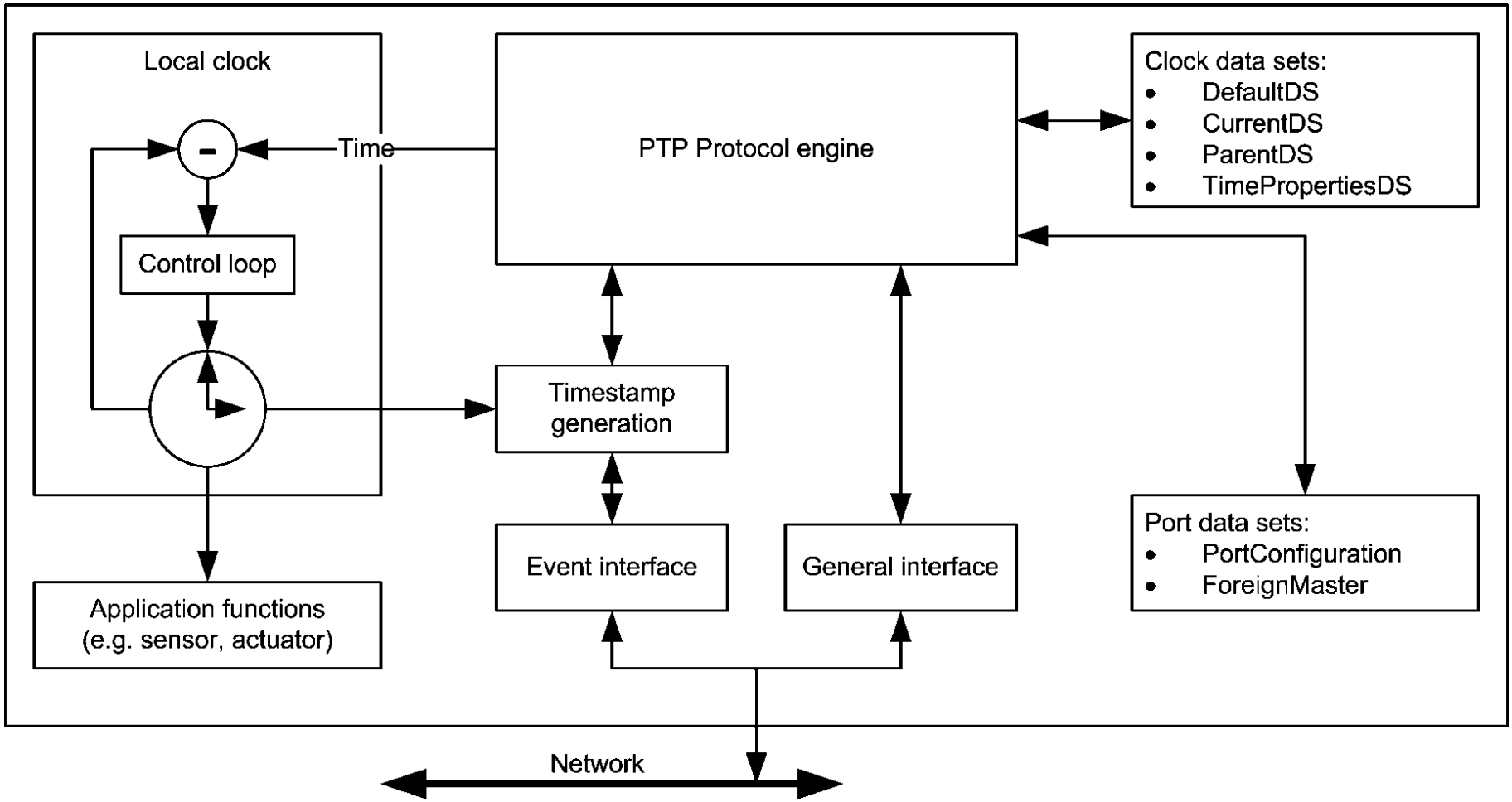


- The nominal increment is choosen according to the nominal oscillator frequency
- The drift is compensated by slightly increasing or decreasing the increment
- A high order portion of the seconds field can be held in software, as long as two-step operation is applied

Zurich University of Applied Sciences

4c) Ordinary Clock Architecture

Implementing IEEE 1588
Model of an Ordinary Clock

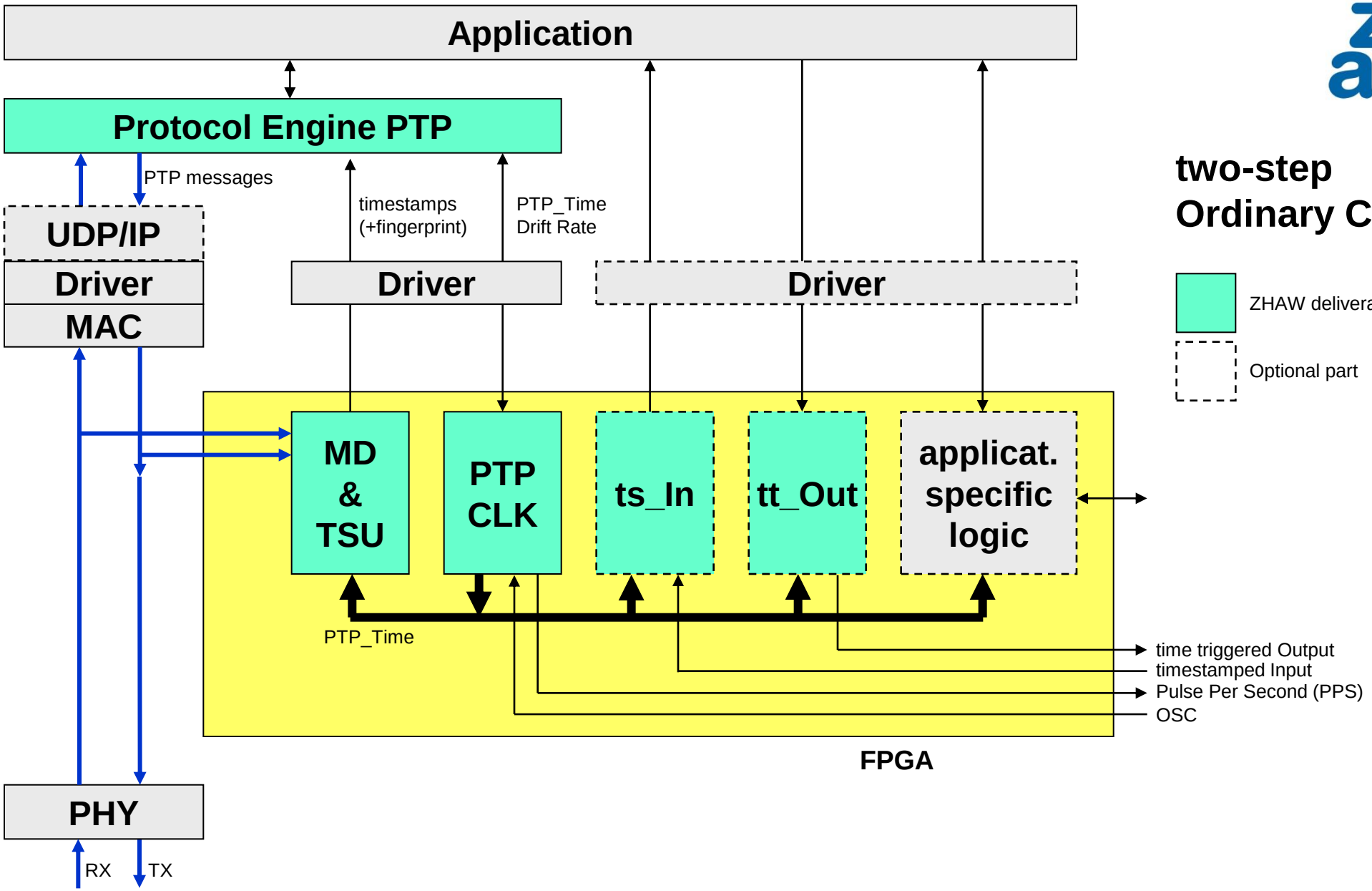


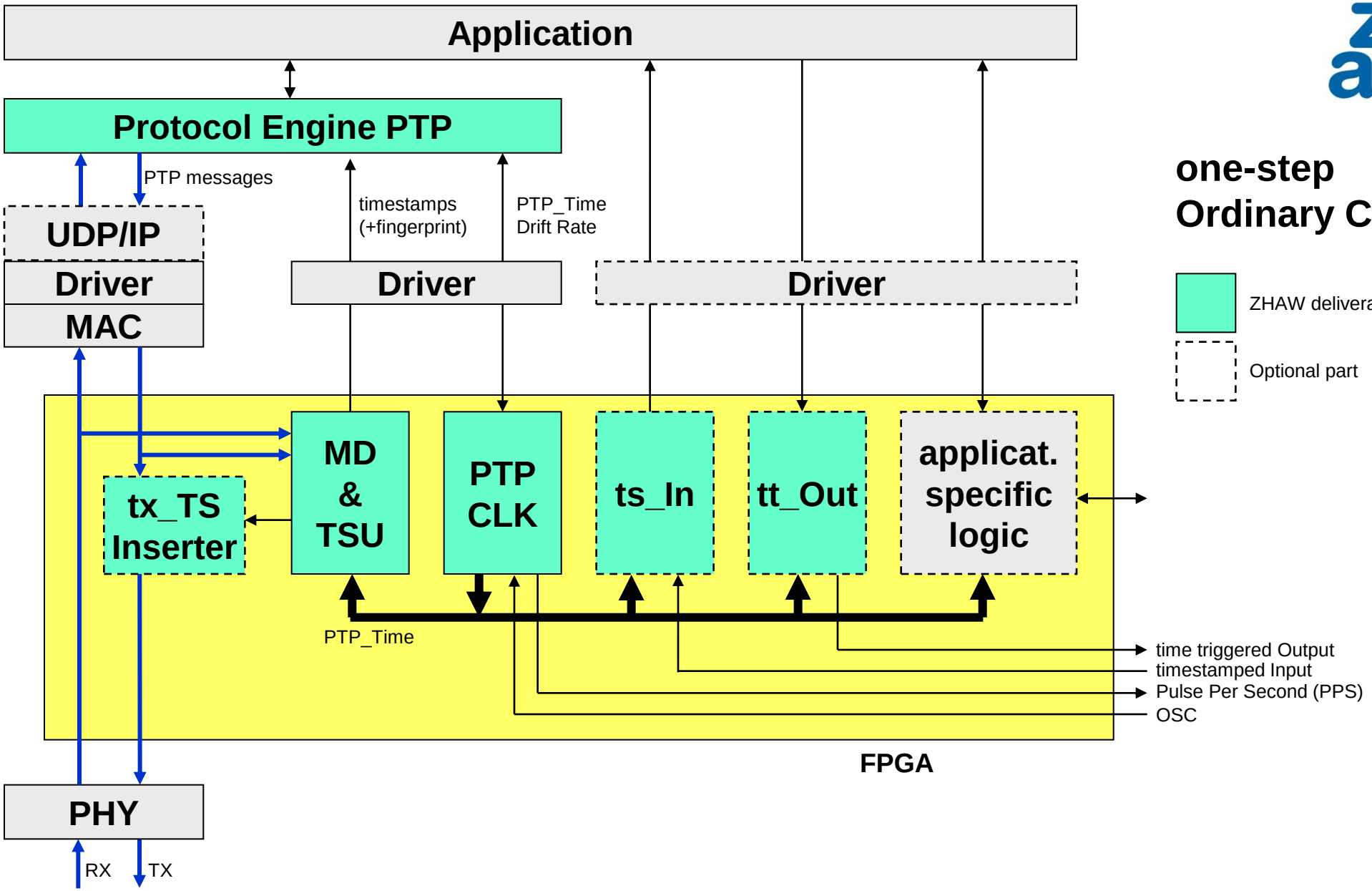
Source: IEEE 1588-2008, Figure 2

Implementing IEEE 1588
ZHAW's IP for Hardware Assistance

Zurich University of Applied Sciences

- Compliant with IEEE 1588 PTPv2 (and PTPv1)
- Both two-step and one-step operation
  - two-step variant taps the MII, RMII, or GMII
  - one-step variant intercepts the MII, RMII, or GMII and allows on-the-fly insertion of timestamp into PTP message (no need for Follow-up message)
- High precision clock with PTPv2 time representation (80 bit)
- Different message mappings supported, such as IPv4, IPv6, and pure Ethernet
- Drift and offset correction mechanisms for clock adjustment
- Various host interfaces available, such as PCI, PC104, MDIO, EBI, serial, and others upon request
- PPS output, programmable time triggered outputs and time stamped inputs
- GPS interface (clock may be synchronised to GPS time source)


Application
Protocol Engine PTP
PTP messages
timestamps
(+fingerprint)
PTP_Time
Drift Rate
UDP/IP
Driver
MAC
Driver
Driver
MD
&
TSU
PTP
CLK
ts_In
tt_Out
applicat.
specific
logic
PTP_Time
time triggered Output
timestamped Input
Pulse Per Second (PPS)
OSC
FPGA
PHY
RX
TX
two-step
Ordinary Clock
ZHAW deliverable
Optional part


Application
Protocol Engine PTP
PTP messages
timestamps
(+fingerprint)
PTP_Time
Drift Rate
UDP/IP
Driver
MAC
Driver
Driver
one-step
Ordinary Clock
ZHAW deliverable
Optional part
MD
&
TSU
PTP
CLK
ts_In
tt_Out
applicat.
specific
logic
tx_TS
Inserter
PTP_Time
time triggered Output
timestamped Input
Pulse Per Second (PPS)
OSC
FPGA
PHY
RX
TX

Implementing IEEE 1588

IP for Hardware Assistance – Functional Blocks

Zurich University of Applied Sciences

- **MD&TSU (Message Detection and Time Stamping Unit)**
  - parses ingress and egress frames
  - generates timestamps for PTP event messages
  - timestamp with corresponding fingerprint is queued to be fetched by the protocol software
- **tx_TS Inserter (for one-step clock only)**
  - transmit timestamps for Sync messages are inserted into the message on-the-fly
  - no need for a Follow_up message
- **PTP CLK (IEEE 1588 clock)**
  - driven by an oscillator that meets application specific stability requirements (free running cheap oscillator up to GPS controlled time source)
  - maintains precise system time in IEEE 1588 representation
  - rate adjusted by varying the increment (drift rate calculated by protocol software)
  - generates a PPS (Pulse Per Second) output signal to observe clock accuracy/stability
- **ts_In (timestamped Input)**
  - generates timestamps for external events
- **tt_Out (time triggered Output)**
  - generates pulses to be used externally (single shot or periodic signal)

Implementing IEEE 1588

IP for Hardware Assistance – Snooping on TX and RX of a Port

Zurich University of Applied Sciences

convert xMII ingress data stream to an octet stream

parse the octet stream

when SFD is detected → read PTP clock

when PTP event message is recognized then

extract UUID, seqNr, msgType from frame
if transmission of frame was successful

then write timestamp, UUID, seqNr, msgType to buffer

convert octet stream to the appropriate xMII egress format

- **Notes:**
  - **This processing can be used for all device types in two-step mode.**
  - **UUID, seqNr, msgType are fingerprint information to allow the software to match a timestamp to the corresponding message.**
    As an alternative the CRC of a message could be used as fingerprint.
  - **Incorrect or incomplete PTP messages should not generate a timestamp.**

Implementing IEEE 1588

IP for Hardware Assistance – Intercepting TX of a Master Port

Zurich University of Applied Sciences

convert xMII ingress data stream to an octet stream

parse the octet stream

when SFD is detected then read PTP clock

when PTP event message is recognized then

extract UUID, seqNr, msgType from frame

if msgType is Sync

then insert the sum of timestamp and TX path delay correction

into the timestamp field of the message

else write timestamp, UUID, seqNr, msgType to buffer

convert octet stream to the appropriate xMII egress format and append new CRC

- **Notes:**
  - **TX path delay correction for timestamps delivered to software is optinal. Be careful in this situation when CRC is used as fingerprint information.**
  - **Delivery of Sync TX timestamp is optional.**

Implementing IEEE 1588
IP for Hardware Assistance – Timestamp Buffering

Zurich University of Applied Sciences

Minimum requirement is a single register per port per direction (may be sufficient if MAC/Driver deliver the timestamp together with the message itself)

- **The software has to read the register before the next PTP event message is transmitted**
  - **a timestamp gets lost otherwise**
- **PTP event messages can arrive back-to-back if the node**
  - **is a unicast master**
  - **is a master serving multiple slaves over a shared medium or a legacy switch**
  - **uses p2p delay measurement messages (note that it may happen in this case that two different PTP messages arrive with the same UUID and seqNr → that's why the msgType may be of interest in the fingerprint information of a timestamp)**
- **The protocol software should be robust enough to cope with**
  - **lost timestamps (i.e. transmitted or received PTP messages without a corresponding timestamp)**
  - **but also with timestamps without a corresponding message**

Implementing IEEE 1588
IP for Hardware Assistance – Timestamp Buffering

Zurich University of Applied Sciences

Using a FIFO to buffer the timestamps allows a more relaxed protocol processing in software

- **For an economic use of resources a common FIFO or a FIFO per direction may be provided**
  - **the software needs to know the port where the timestamp was generated**
- **Reading the FIFO**
  - **the protocol software probably wants to process the timestamps in a different order as they are read out from the FIFO**
  - **reading the FIFO can be triggered by the transmission or reception of a frame or a timer event (PTP protocol engine is executed cyclically)**
- **ZHAW implementation**
  - **the hardware FIFO is copied to a list maintained in software**
  - **the protocol engine consumes the timestamps when needed**
  - **an ageing mechanism removes timestamps which were not consumed**

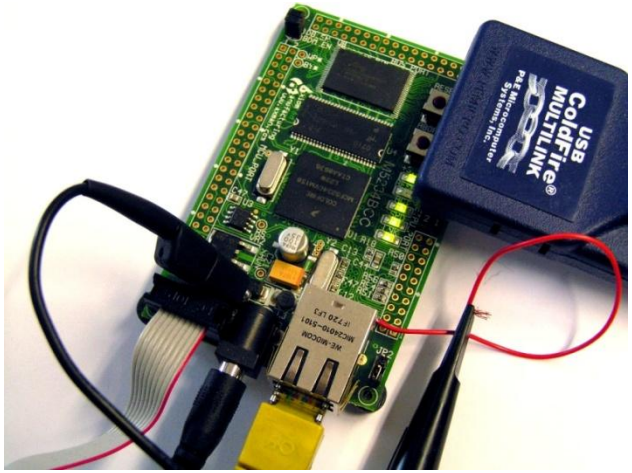
Implementing IEEE 1588

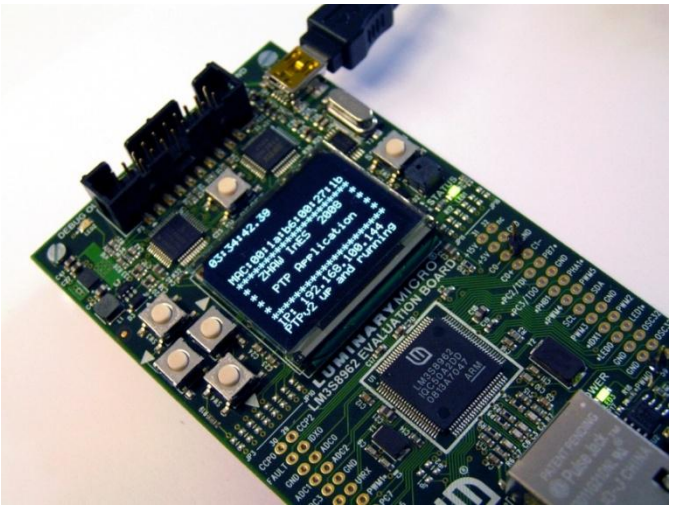
Functions typically implemented in Software

- Execute the Best Master Clock Algorithm and select port status
- Transmit and receive PTP messages
  - at selected rate
  - reception supervised by timeout
- Acquire timestamps and respective fingerprint information
- Calculate drift and offset
- Control the PTP clock (set time or correct offset, set drift)
- Execute filtering algorithms
- Execute servo loop control (PI)
- Monitor
  - behaviour of master
  - switchover to other master
  - accuracy of local clock (calculate variance)
- Manage special situations
  - Leap seconds
  - Loss of master (holdover)
- Provide an API allowing the use of PTP time (external events and triggers)
- Provide a management interface
- etc.

Implementing IEEE 1588

Various IEEE 1588 Evaluation Kits from ZHAW





Zurich University of Applied Sciences

4d) Boundary Clock Architecture

The Boundary Clock is a clock bridge

Implementing IEEE 1588
Protocol Software for Boundary Clock

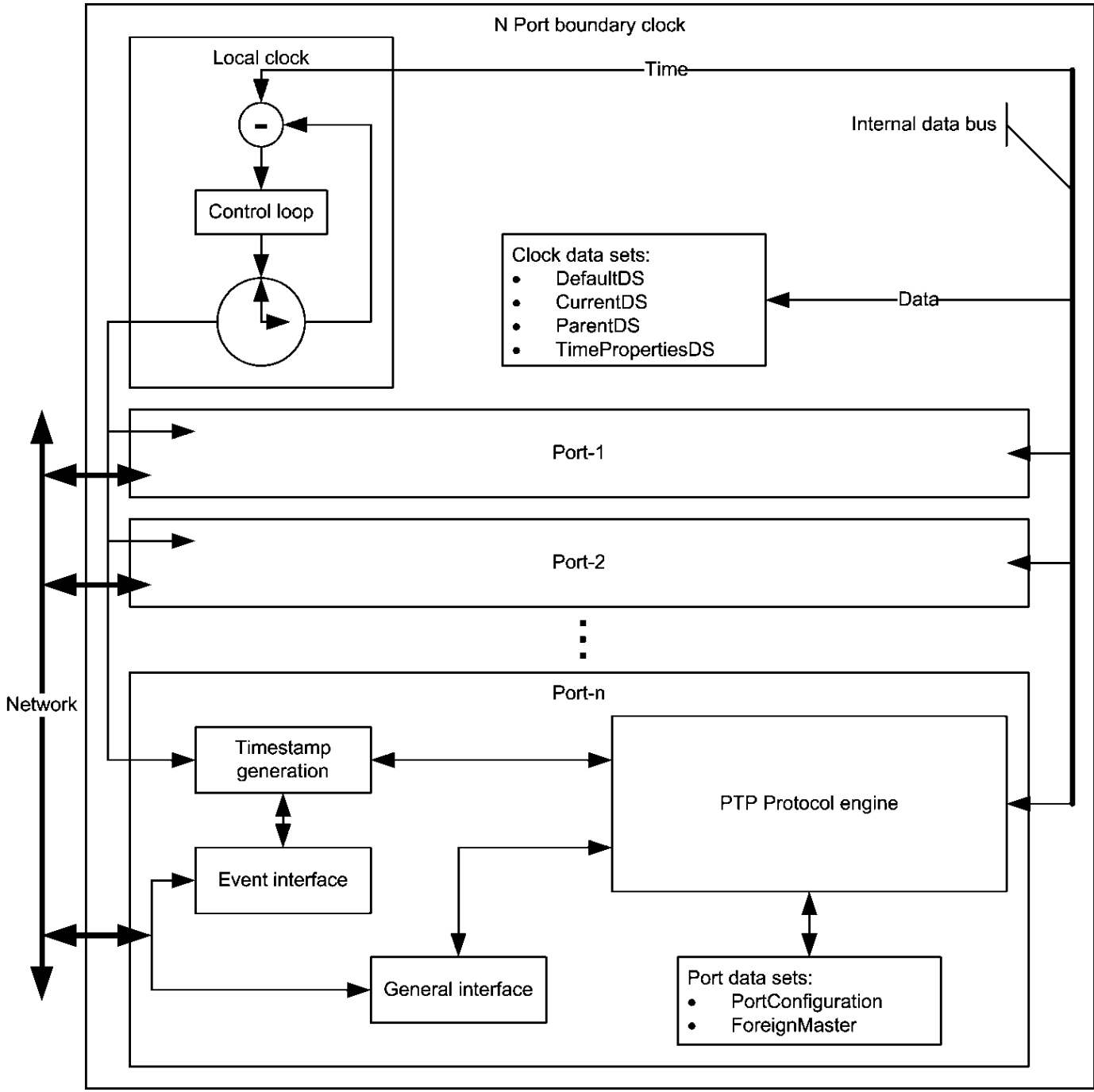
Zurich University of Applied Sciences

Protocol software is the same for OC and BC

- **Both have to maintain and control one single PTP clock**
- **While the OC has one single port, a BC has *n* of them**
  - **Each port has ist own protocol engine**
  - **If the protocol software is written in an object oriented style, a BC has *n* instantiations of the port object**

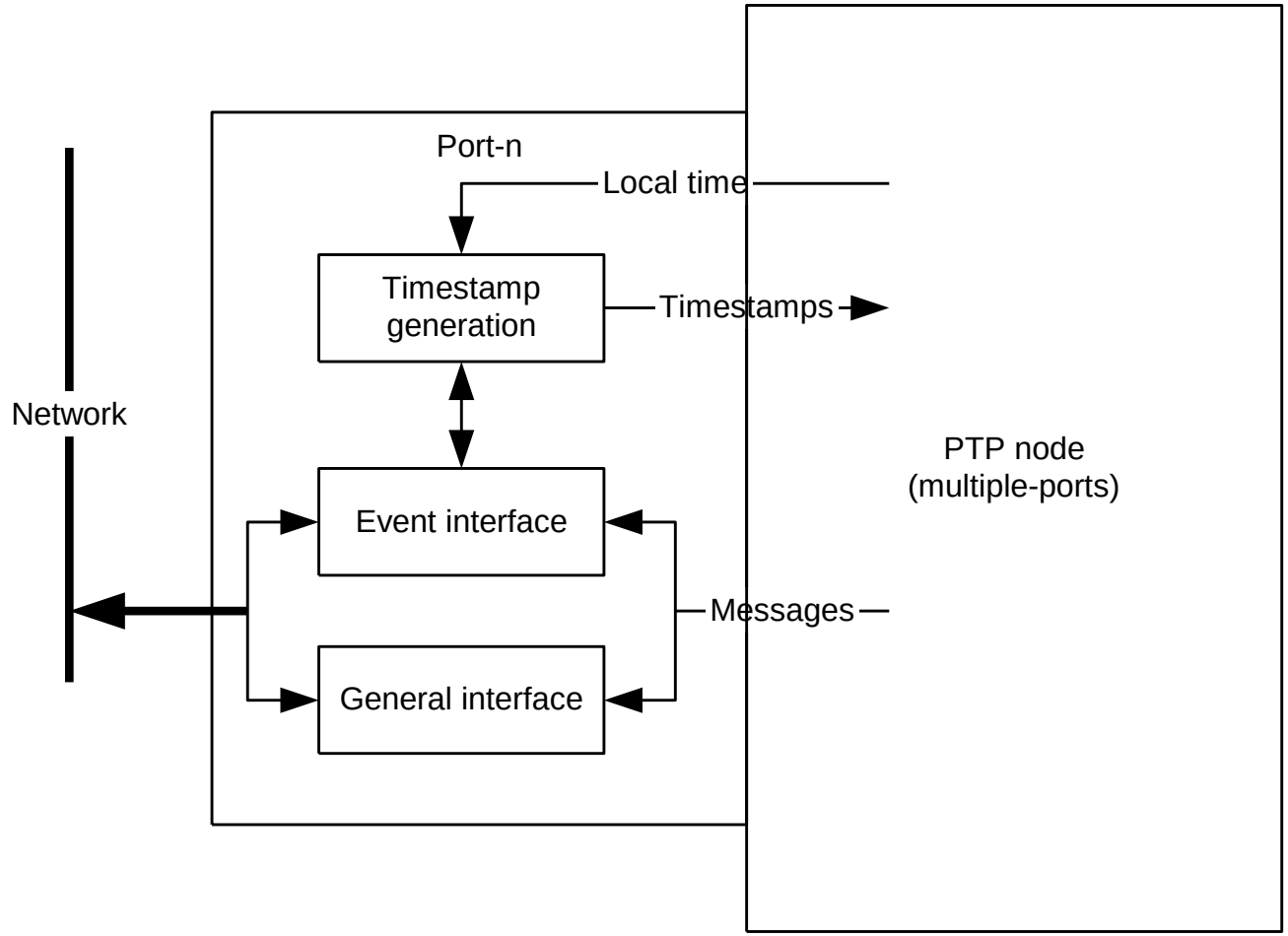
Implementing IEEE 1588

Model of a Boundary Clock



Source:
IEEE 1588-2008, Figure 3

Implementing IEEE 1588
Port Model



Source: IEEE 1588-2008, Figure 21

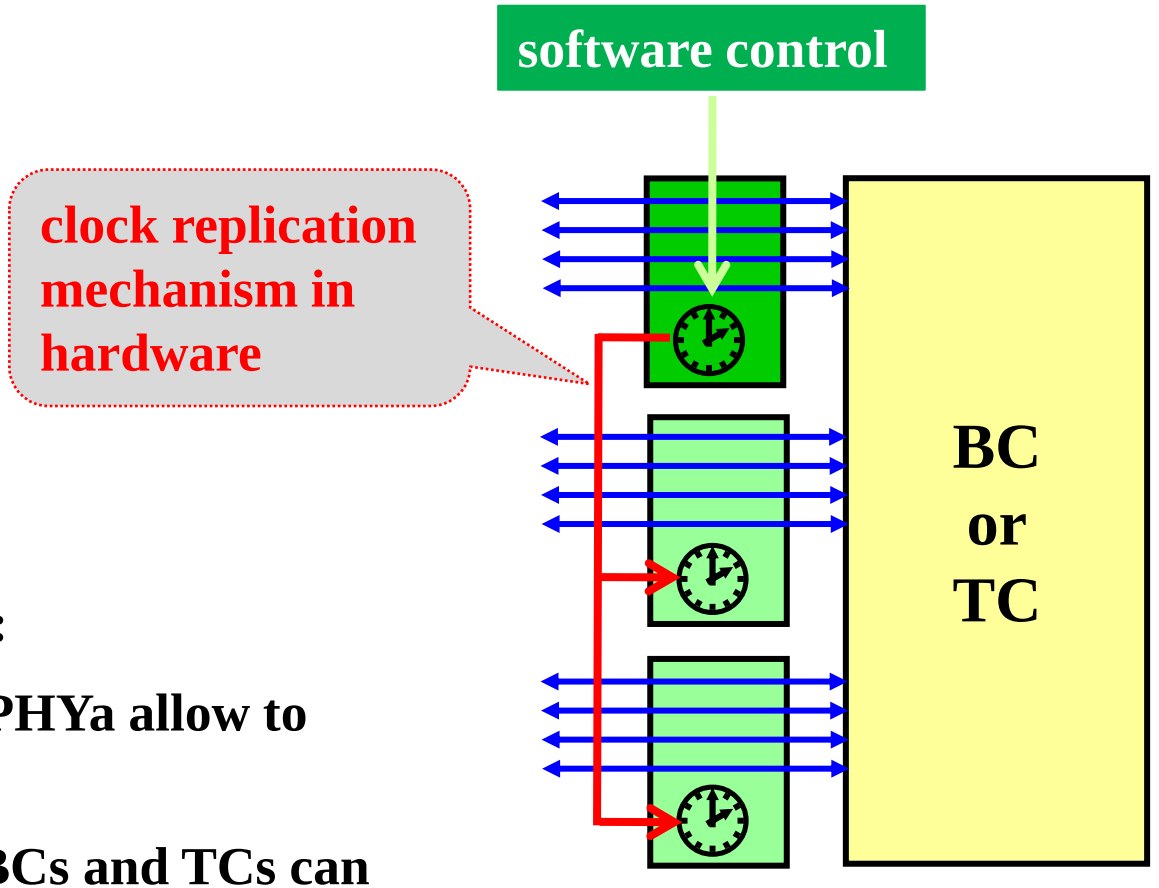
Implementing IEEE 1588
Scaleable Multi-Chip Hardware Assistance

Zurich University of Applied Sciences

- For scalability reasons it may be necessary to split the IEEE 1588 hardware assistance of a node into several components
- PTP messages need to be timestamped on the basis of the same timescale on all ports of the node
- Providing a common timebase can be achieved by
  1. Controlling a local PTP clock in every device
     - requires a hardware triggered start-up (set all PTP clocks and start them simultaneously)
     - all PTP clock have to be controlled identically
     - all PTP clock need a common frequency source
  2. Controlling one single PTP clock and replicate it to the other components at a high rate
     - high accuracy may be achieved even if each component has its own local frequency source
     - consistency can easily be guaranteed

Implementing IEEE 1588

Multi-Chip Hardware Assistance based on Clock Replication



Example for clock replication:

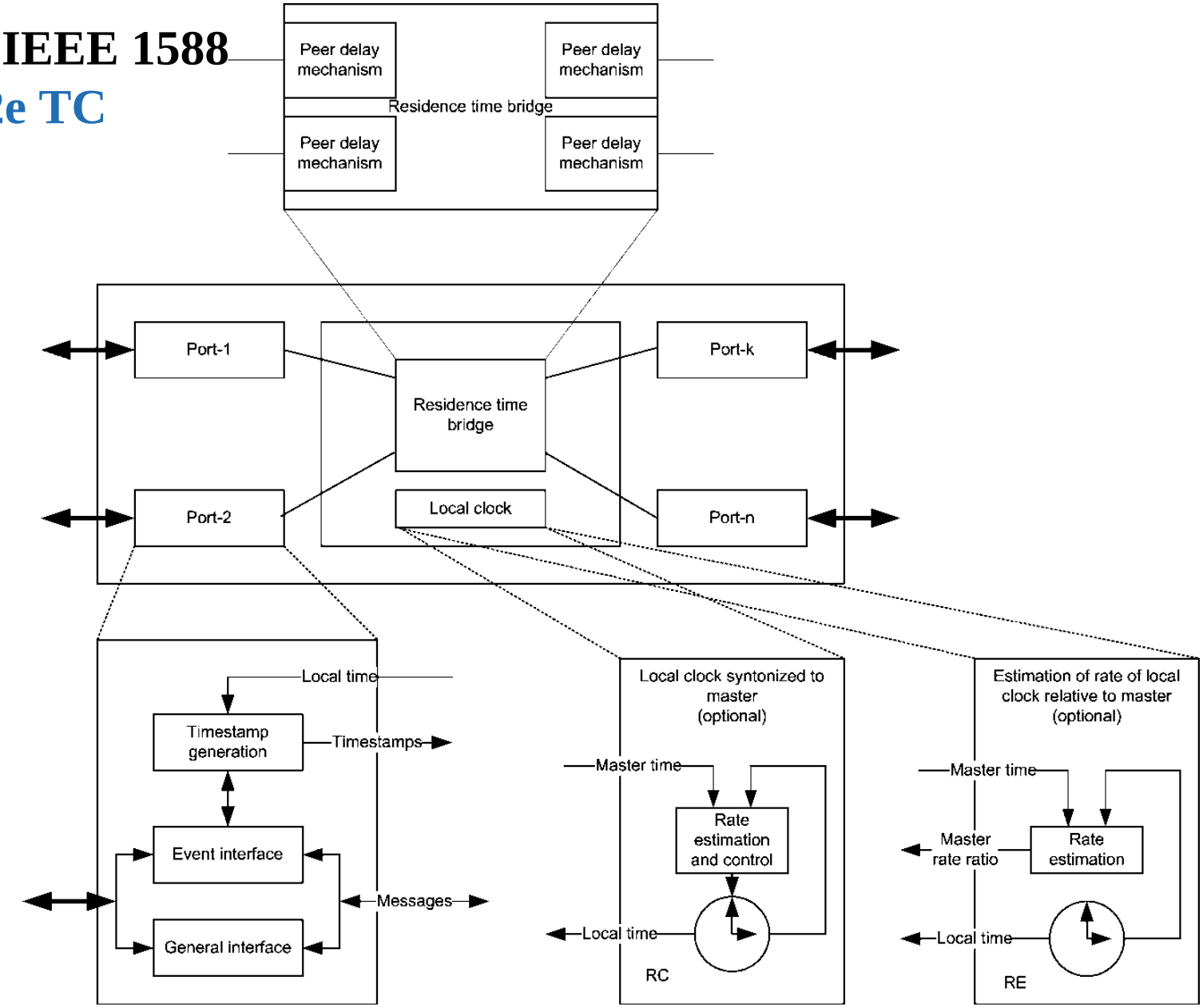
- **The Renesas PTP capable PHYa allow to replicate PTP time**
- **Multiport devices such as BCs and TCs can be implemented easily on the basis of this feature**

Zurich University
of Applied Sciences

4e) Transparent Clock Architecture

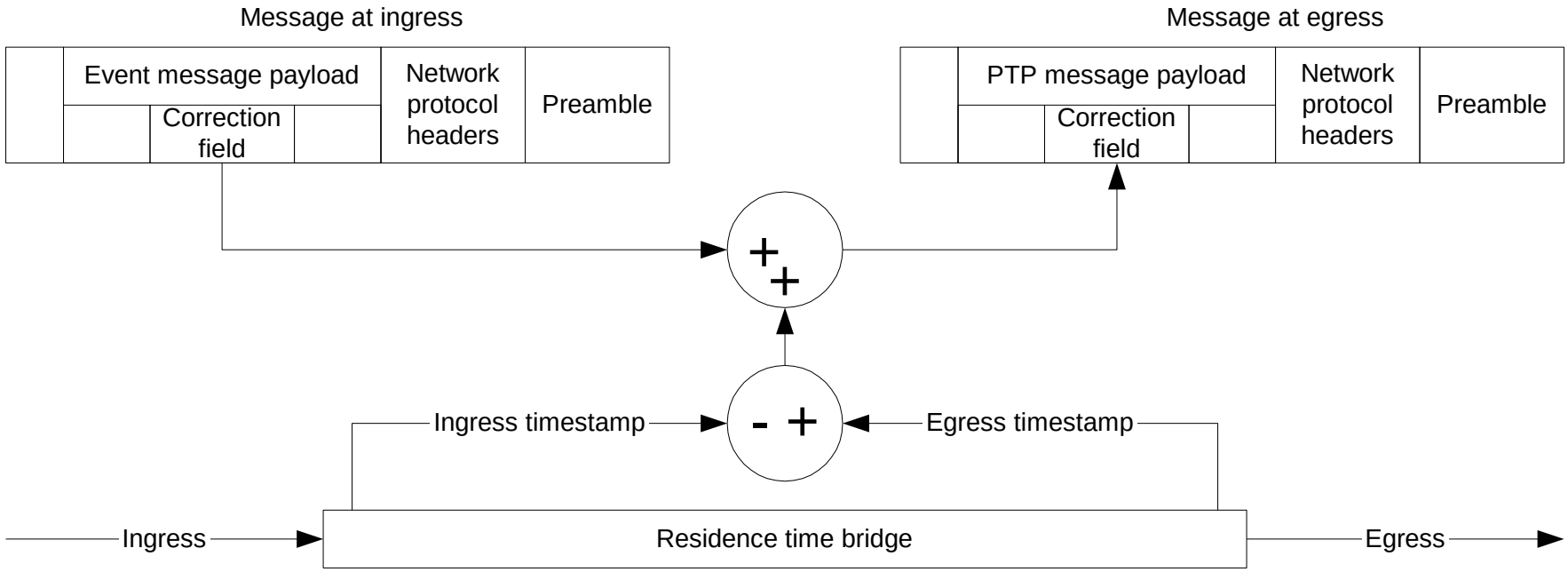
Implementing IEEE 1588

Model of an e2e TC



Source: IEEE 1588-2008, Figure 4

Implementing IEEE 1588
End-to-end Residence Time Correction Model



Source: IEEE 1588-2008, Figure 5

Implementing IEEE 1588
Measurement of Residence Time

Zurich University of Applied Sciences

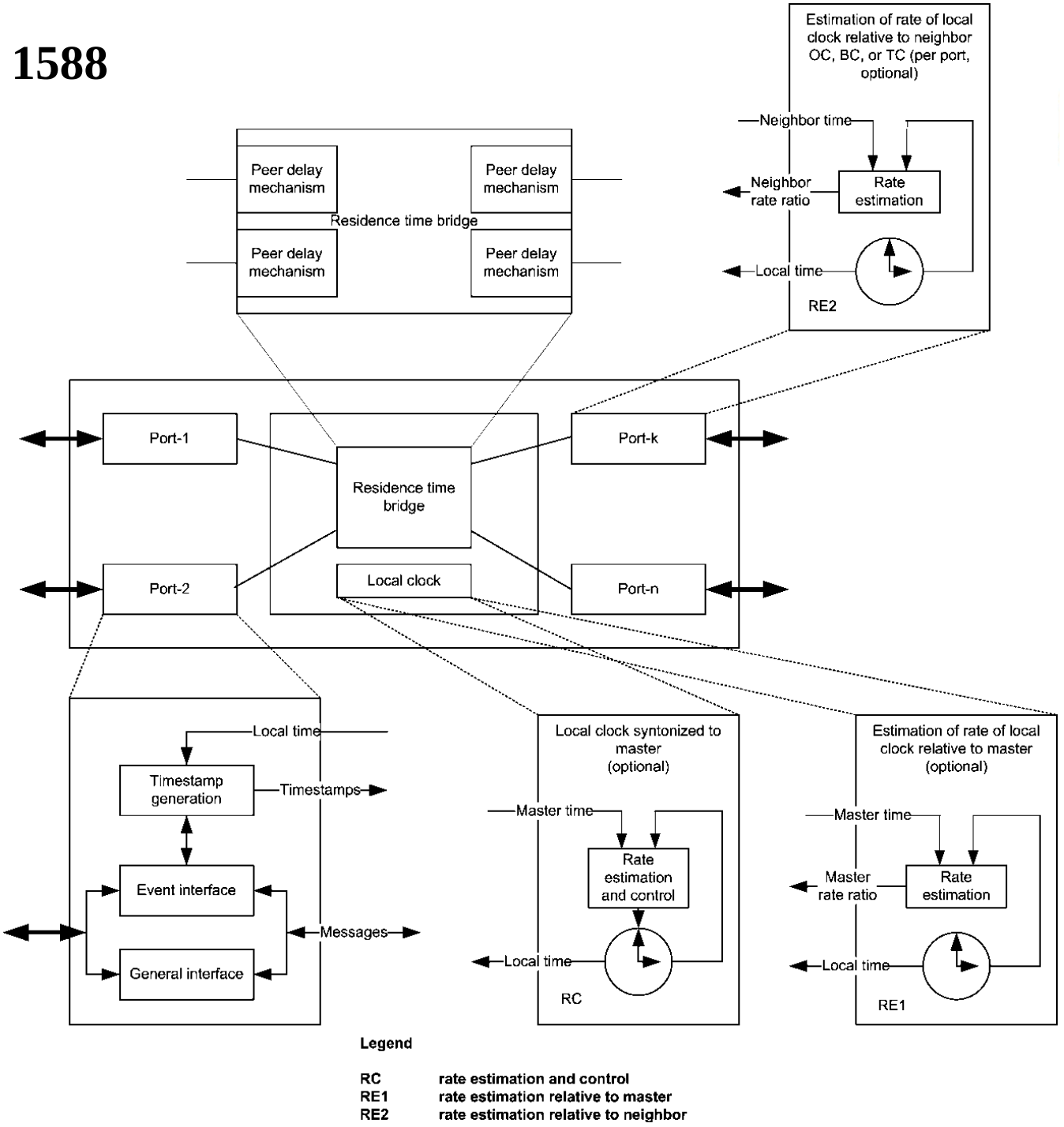
- A TC does not need to be synchronized → it is sufficinet if time intervals can be measured precise enough

Two methods can be applied

- Syntonization of the TC, i.e. adjust the local clock to the same rate as the master clock
  - the propagation of the reference rate takes time (control loop)
- Rate difference between local clock and master can be estimated and taken into account
  - faster propagation
- Both methods are optional, but
  - the longer the residence time can be the more important is the precision of the local rate (e.g. 1 ms residence time measured with 50 ppm → 50 ns)
  - PTP messages in two-step TCs are forwarded by the CPU

Implementing IEEE 1588

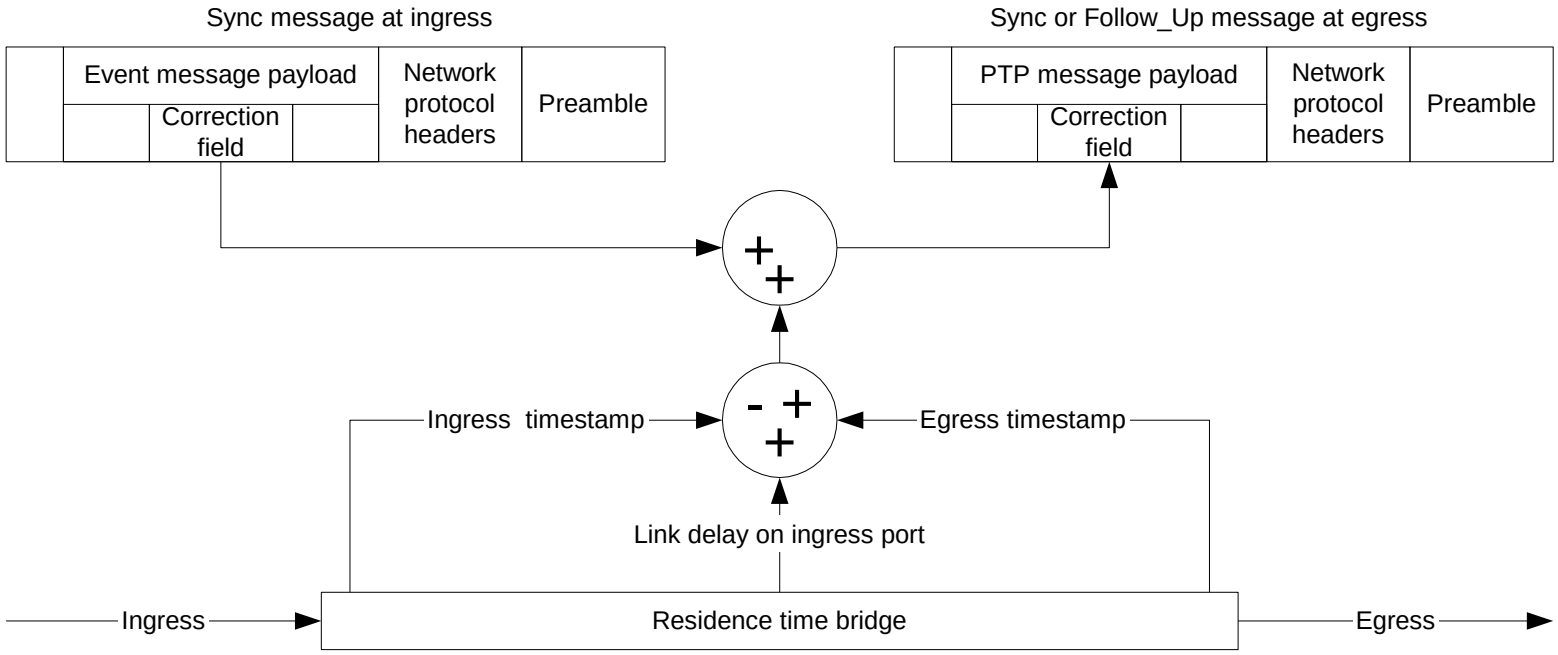
Model of a p2p TC



Source:
IEEE 1588-2008, Figure 7

Implementing IEEE 1588

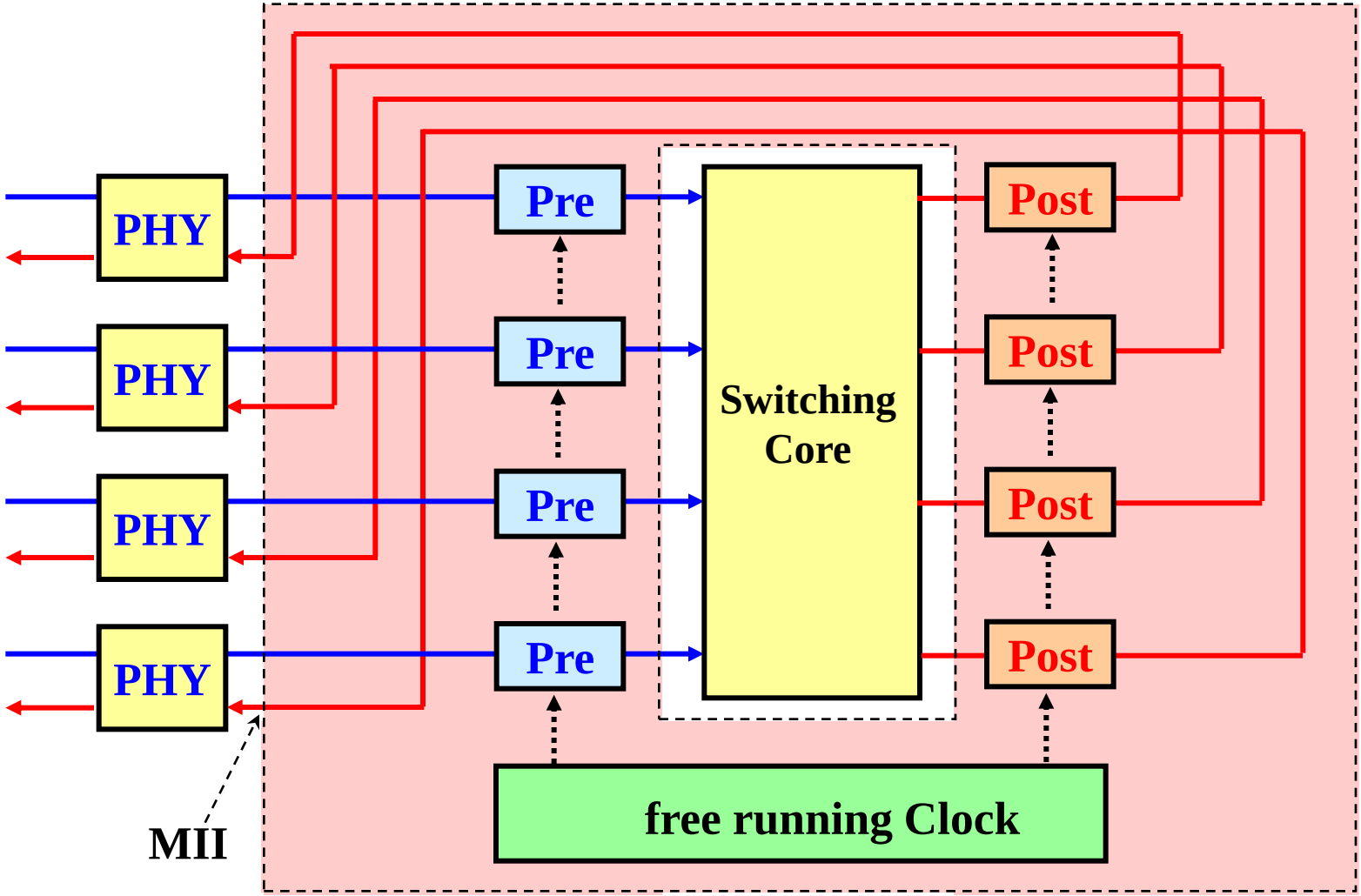
Peer-to-Peer Residence Time and Link Delay Correction Model



Source: IEEE 1588-2008, Figure 8

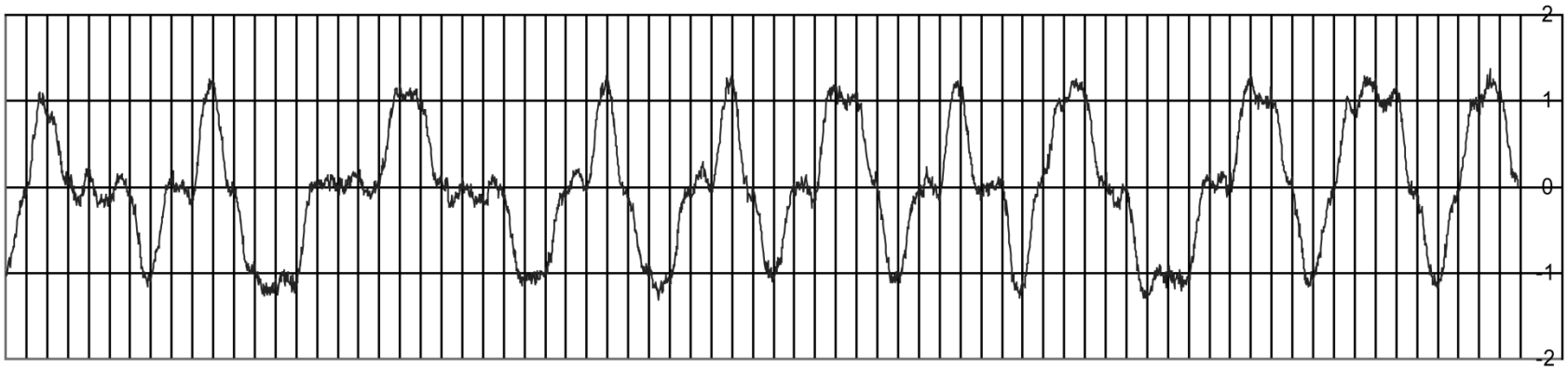
Implementing IEEE 1588

One-step Transparent Clock – Implementation



Zurich University
of Applied Sciences

5) Synchronous Ethernet (SyncE)

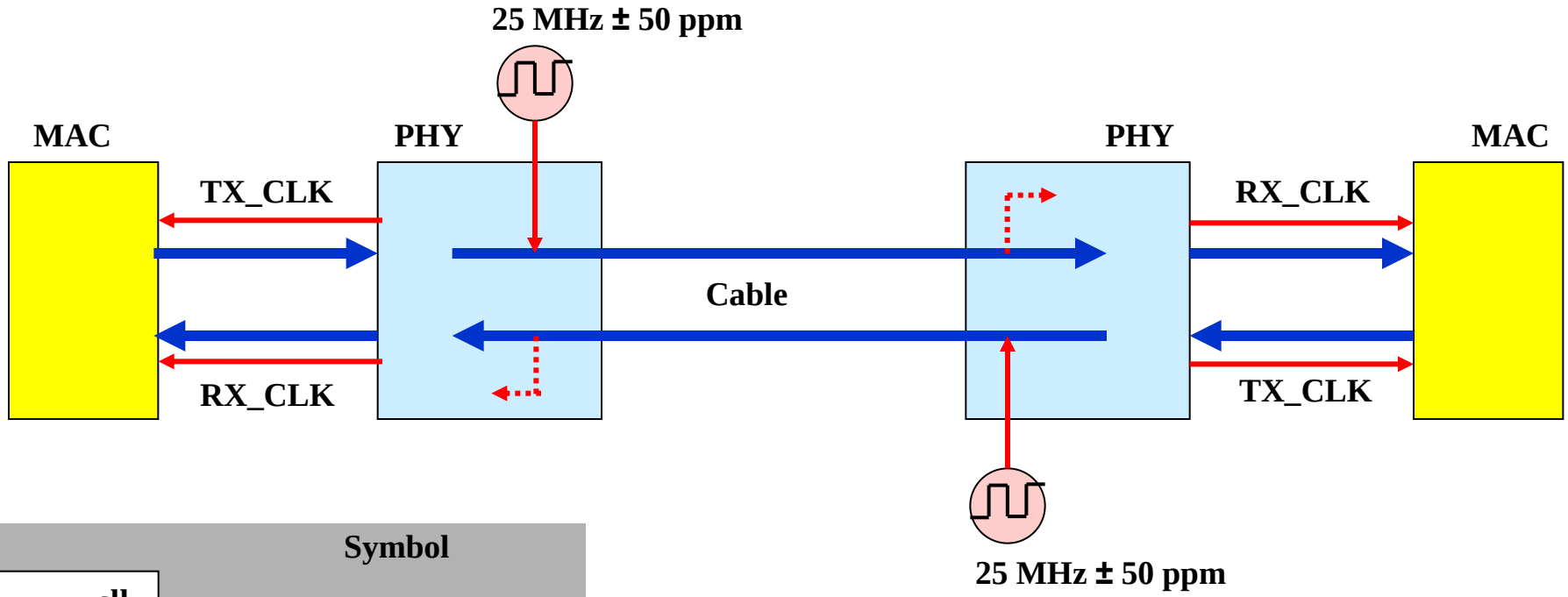


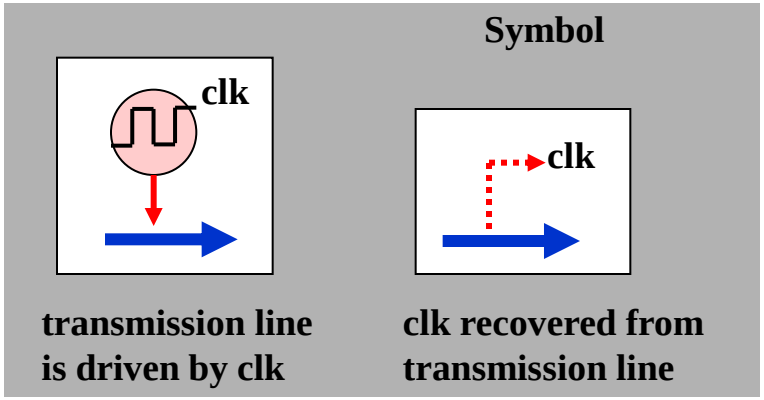
Synchronous Ethernet
Physical Layer Timing in Legacy Ethernet

- Ethernet works perfectly well with relatively inaccurate clocks
- In general each Ethernet transmitter may use its own clock
  - nominal clock rate is the same, but deviations (e.g. ± 50 ppm for 10/100/1000 Mbps Ethernet) are allowed
  - the dimensioning is such that physical layer buffers do not underflow or overflow because of clock deviations
- Details differ according to transmission technology
  - where the two directions of a link use different media (i.e. separate wire pairs or separate fibers), both directions may have independent clocks
  - GBE over twisted pair uses all wire pairs simultaneously in both directions
    → signal processing (echo compensation technique) requires same clock on both directions of a link
    → one PHY acts as the master, the other as slave

Synchronous Ethernet

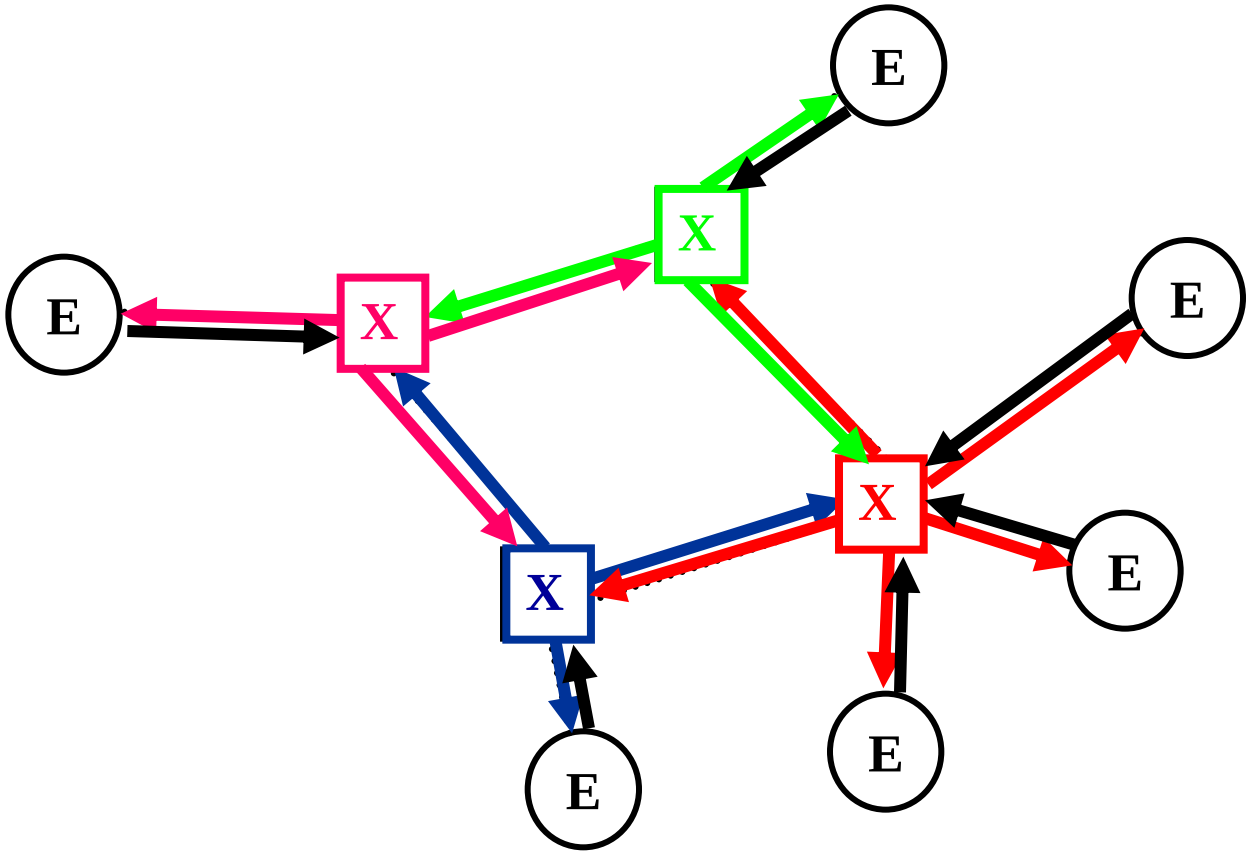
Timing of a Fast Ethernet Link (100 Base-TX)





Synchronous Ethernet

Physical Layer Timing in Legacy Ethernet



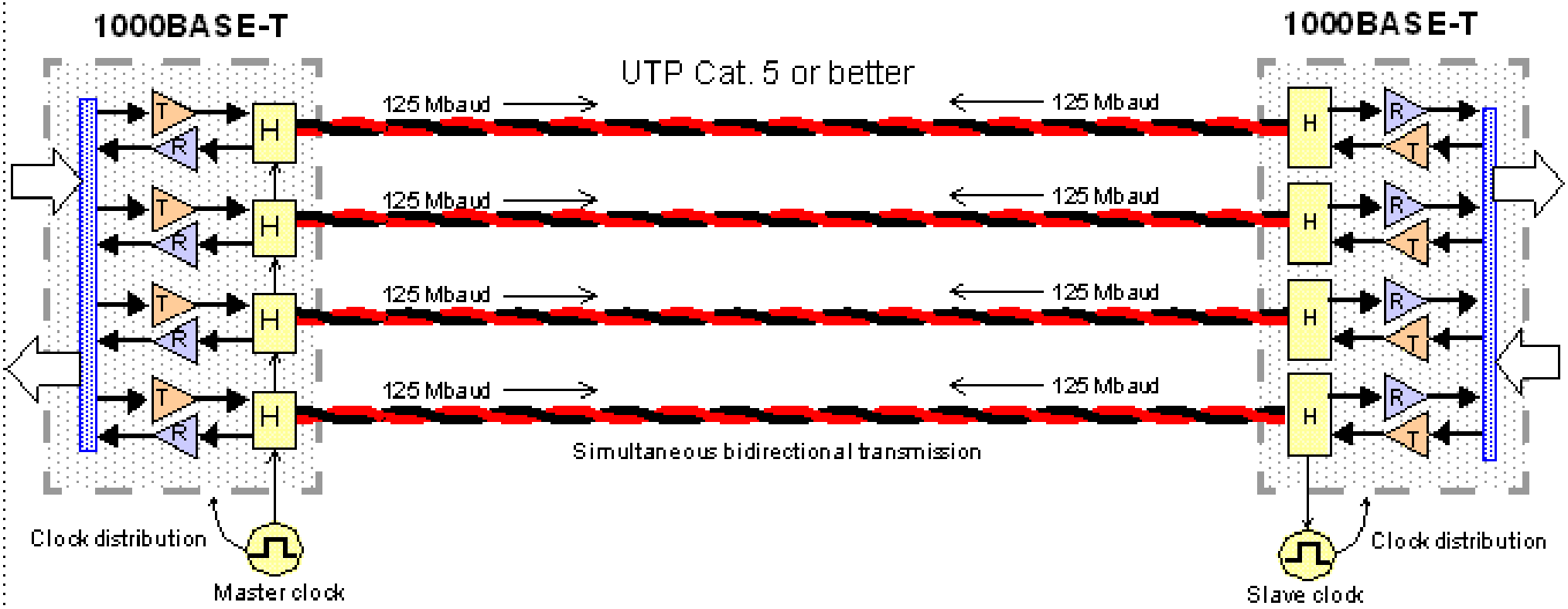
Synchronous Ethernet
Timing of a Gigabit Ethernet Link (1000 Base-T)

Zurich University of Applied Sciences

- 1000 Base-T transmission is split on 4 wire pairs operation simultaneously in both directions
  - transmitter and receiver are coupled via a hybrid
  - echo compensation is applied
  - both directions require the same clock
- A 1000 Base-T PHY can operate as master or as slave.
- Master/slave role selection is part of the auto-negotiation procedure.
- A prioritization scheme determines which device will be the master and which will be slave.
- The supplement to Std 802.3ab, 1999 Edition defines a resolution function to handle any conflicts:
  - multiport devices have higher priority to become master than single port devices.
  - if both devices are multiport devices, the one with higher seed bits becomes the master.

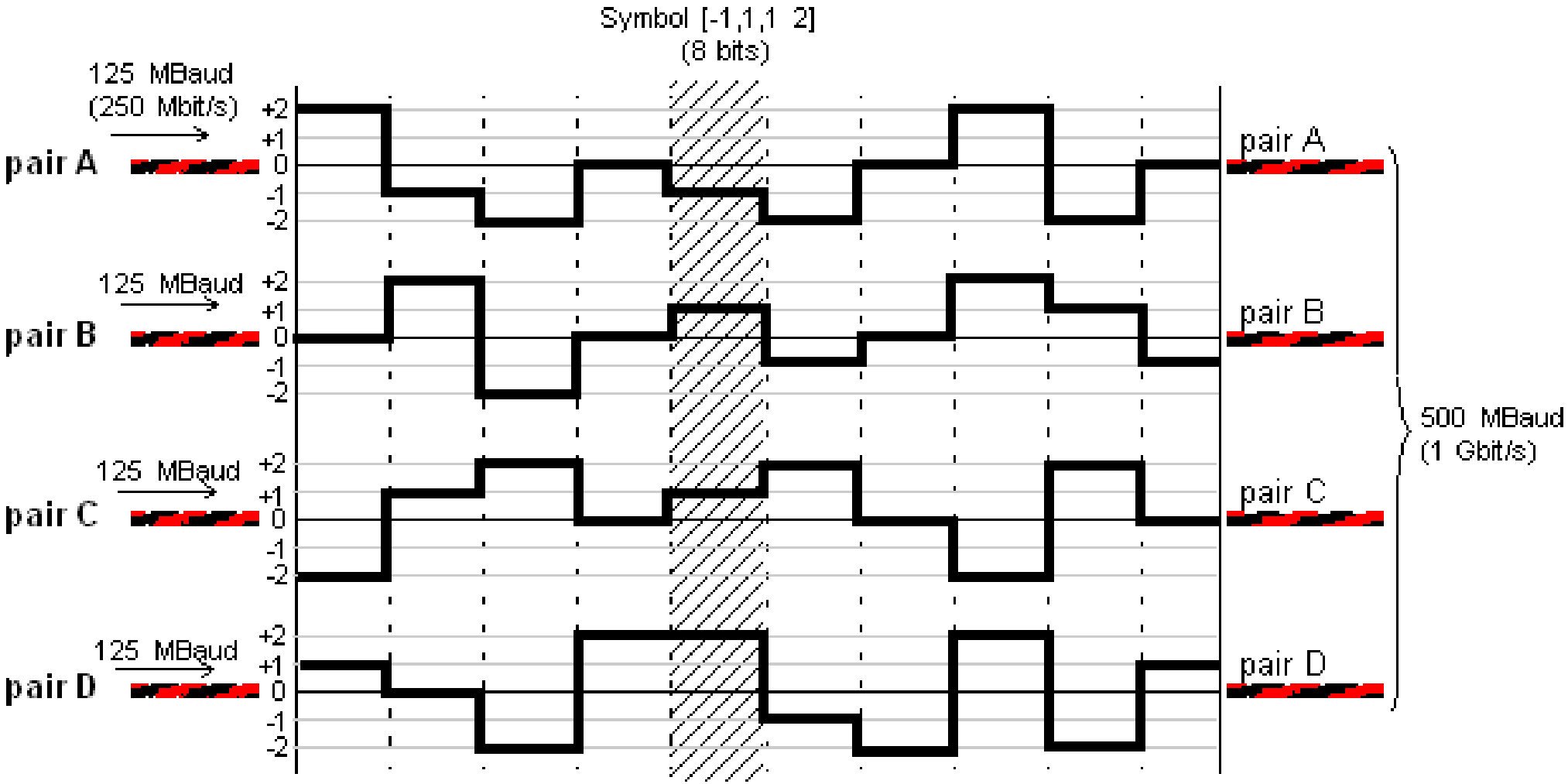
Synchronous Ethernet

1000 Base-T uses 4 pairs simultaneously in both directions



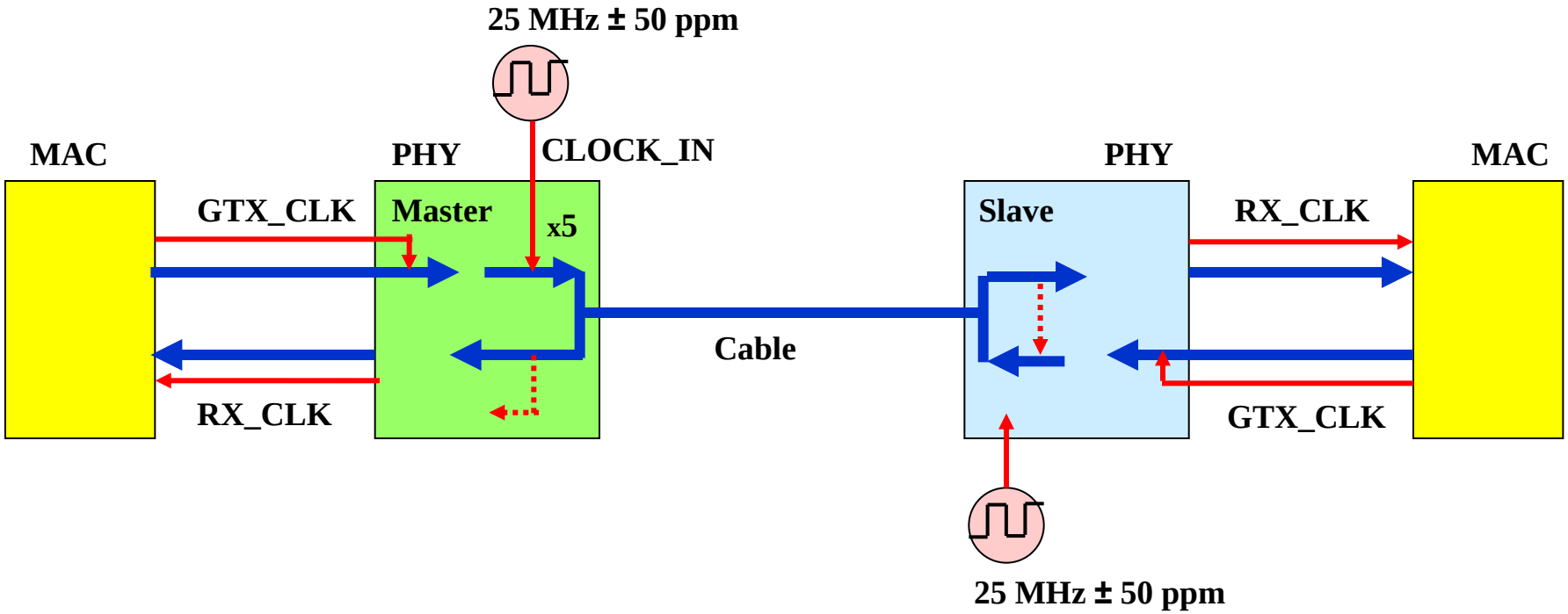
Synchronous Ethernet

1000 Base-T Pysical Layer Signalling with Echo Compensation



Synchronous Ethernet

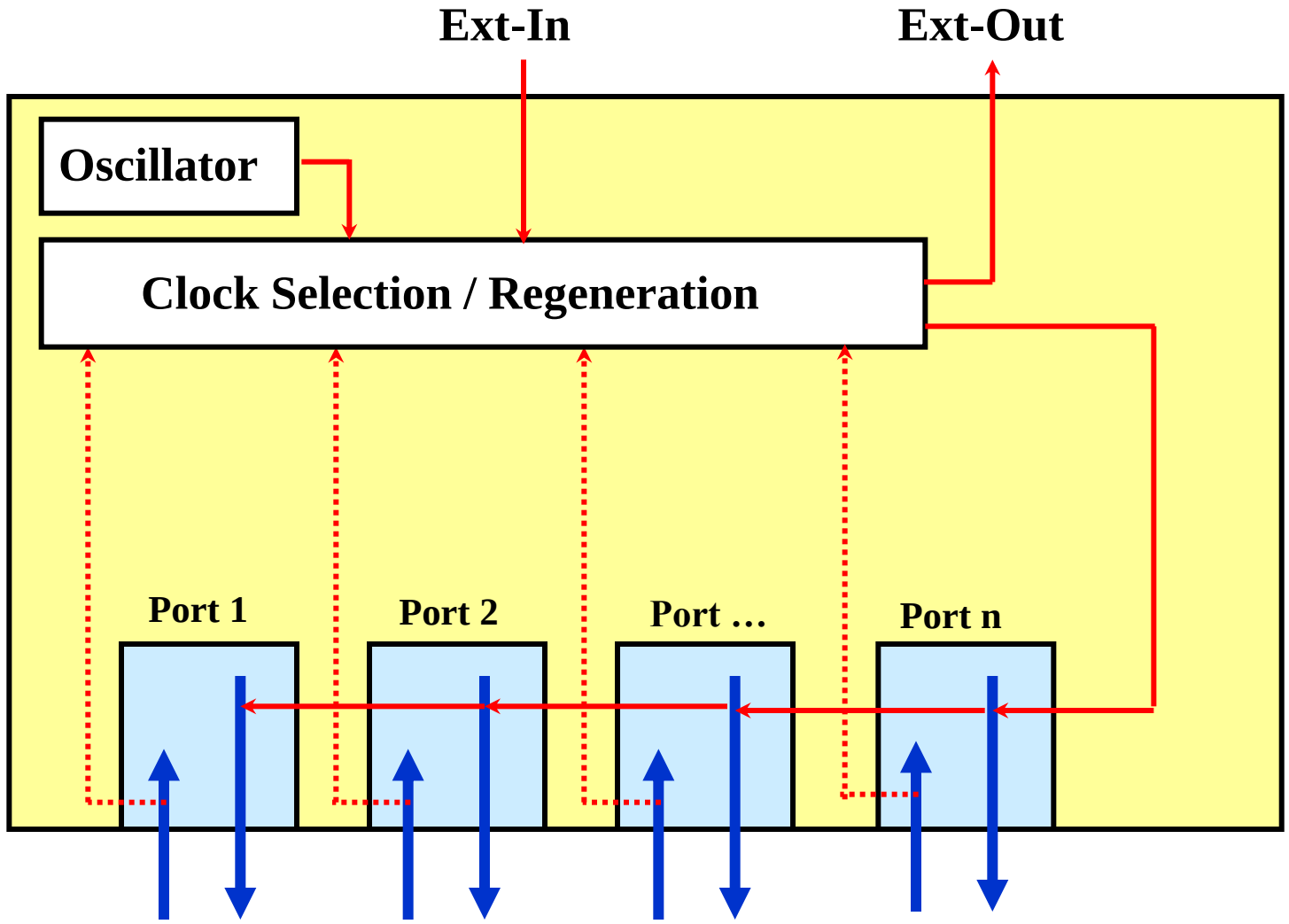
Timing of a Gigabit Ethernet Link (1000Base-T)



- The Master PHY uses the internal 125 MHz clock generated from CLOCK_IN to transmit data on the 4 wire pairs.
- The Slave PHY uses the clock recovered from the opposite PHY as the transmit clock.

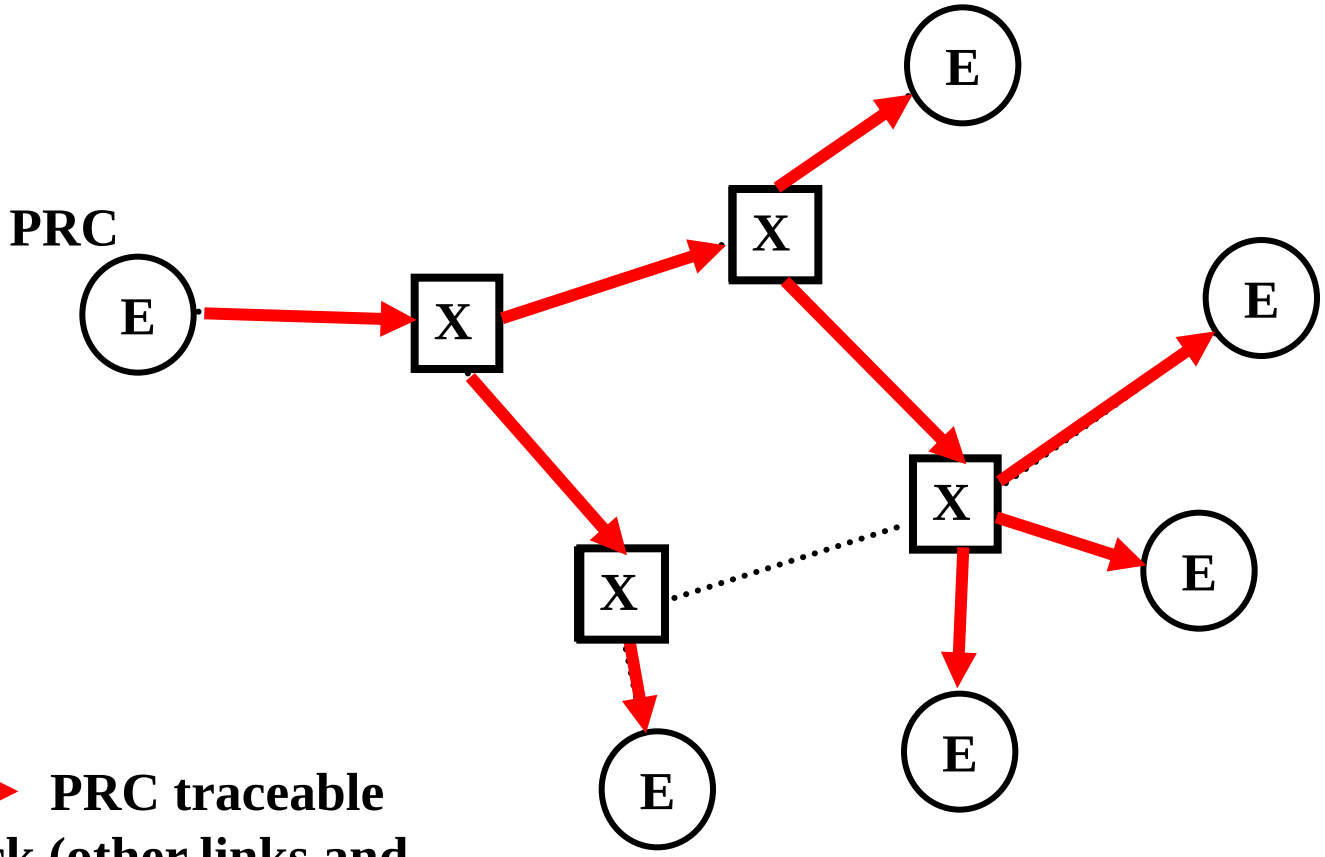
Synchronous Ethernet

Clock Sources for a Synchronous Ethernet Switch



Synchronous Ethernet

Physical Layer Timing in Synchronous Ethernet



PRC traceable clock (other links and directions may be free running)

Synchronous Ethernet
Concept

Zurich University of Applied Sciences

- The concept of SyncE has been proposed, elaborated, and standardized by the Telecom community in ITU-T by applying the traditional SDH/SONET clock distribution concept to Ethernet networks
- The Primary Reference Clock (PRC) frequency is distributed on the physical layer
  - a receiver can lock to the transmitter‘s frequency
  - a switch selects the best available clock to be used as transmission clock
  - this results in a hierarchical clock distribution tree
- OAM messages (Synchronization Status Messages) are used to signal clock quality and sync failure conditions of the upstream switch
  - to allow selection of the best available timing source (stratum of upstream source)
  - to avoid timing loops
  - OAM messages are an equivalent of the SSM field in the SDH/SONET frame overhead

Synchronous Ethernet
Concept

Zurich University of Applied Sciences

- The active layer 2 data forwarding topology (as established by spanning tree protocol) and clock distribution tree are independent (i.e. a blocked port can deliver the clock to its neighboring switch)
- Design rules (topology restrictions, priorities for source selection) guarantee clock quality
- Clocking of Ethernet devices is changed in a way that is fully conforming with IEEE 802.3 standards
- Standard PHY chips can be used as long as a few conditions are met, e.g.
  - PHY provides the recovered receive clock to the external world
  - GBE PHY allows master/slave role to be set by software (no automatic selection)

Synchronous Ethernet
Compared with IEEE 1588

Zurich University of Applied Sciences

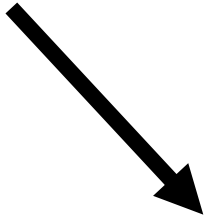
zh aw

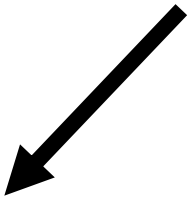
Synchronous Ethernet

- Clock distribution based on Ethernet‘s physical layer
- Provides frequency only
- Performance is independent of data traffic

IEEE 1588

- Application layer protocol with hardware assistance
- Provides frequency and time of day
- May be susceptible to specific data traffic patterns





Complementary technologies, can be used in combination:

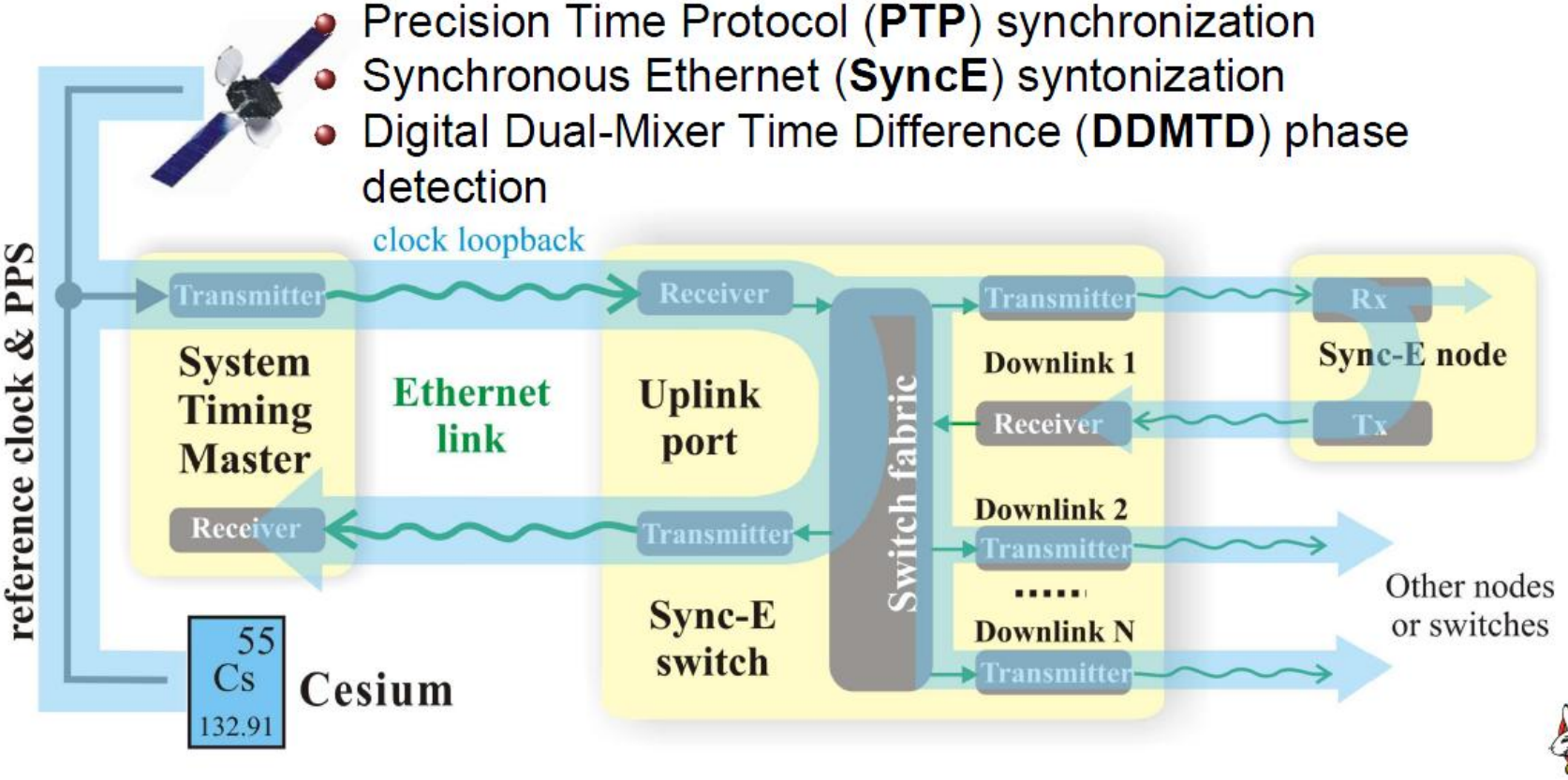
Syncronous Ethernet delivers accurate and stable frequency to all nodes while IEEE 1588 can deliver phase alignment and time of day, where required.

Synchronous Ethernet
Combined with IEEE 1588

Zurich University of Applied Sciences

White Rabbit, CERN's PTP-based control and timing system

- Synchronization with **sub-ns** accuracy and **ps** precision
- Combination of
  - Precision Time Protocol (**PTP**) synchronization
  - Synchronous Ethernet (**SyncE**) syntonization
  - Digital Dual-Mixer Time Difference (**DDMTD**) phase detection



Source: Maciej Lipinski, CERN

Many thanks for your attention!

hans.weibel@zhaw.ch

Zurich University of Applied Sciences
Institute of Embedded Systems
http://ines.zhaw.ch/ieee1588