Standard Ethernet for Real-Time Industrial Application

IEEE1588 Precision Time Protocol

T3LAB Bologna -13 Dicembre 2012
Agenda

- Industrial Ethernet
- Standard Ethernet in real-time industrial applications
- Switched Ethernet
- IEEE 1588 PTP Protocol
- IEEE 1588 PTP Implementation issues
- Virtual LAN applications
- PTP Network Demo
Industrial Ethernet (IE)

- Use of the Ethernet family of computer network technologies in an industrial environment, for automation and process control
- Adapt Ethernet for the needs of industrial processes, which require real time behavior
- Use of standard router, switches, hub, ecc..
- Interoperability
- Innovation: high data rate up to 1Gbit/s
Industrial Ethernet: Fieldbus Overview

- Profinet
- Ethernet/IP
- Sercos III
- Powerlink (B&R)
- Modbus TCP
- EtherCAT (BeckHOFF)
- Switched Ethernet
- Plain Ethernet
Industrial Ethernet structures

- Non-Real-Time
  - Non Real – Time protocol
  - TCP / UDP / IP

- Top of TCP/IP
  - Real – Time protocol
  - TCP / UDP / IP

- Top of Ethernet
  - Real – Time protocol
  - Ethernet

- Modified Ethernet
  - Real – Time protocol
  - Modified Ethernet

- Universal cabling
Industrial Ethernet applications

IE Fieldbus

Factory automation

Motion Control

„Component based Automation“

Soft Real Time (Software Based)

Isochronous Real Time (Hardware Based)
IE – World Market (IMS Research)

Strength of Ethernet TCP/IP, the 'standard' form of Ethernet

World market shares for Industrial Ethernet components
Standard & Switched Ethernet

- TCP/IP Protocol
- PTP Protocol
- STP/RSTP Protocol
- VLAN Support
- Gigabit Ethernet
- **Specialized Ethernet HW**
  - Managed switches
- Redundant topologies
- Power over Ethernet (PoE)
Switched Ethernet architecture

- Predictable performance
- Standardization
- QoS feature
- Broadcast & Multicast
- Network analyzers
- Latency increase with number of cascading nodes

- IEEE 1588 HW timestamp
- RSTP support
- Gigabit Ethernet
- Security with VLAN
- Diagnostics (port status, amount of traffic)
IEEE 1588: Standard for a Precision Clock Synchronization Protocol

Distribution of frequency and time over a packet network
Time synchronization in distributed system

- To coordinate measurement instant (sampling, triggering)
- To measure time intervals (and to calculate derived quantities)
- As a reference to determine the order of events
- As a basis for the execution of coordinated actions (time based behavior)
  - Scheduled execution of scripts
  - Coordinate actuators
Time synchronization is required

A Measurement and Control System Supporting IEEE 1588
Why is necessary?

- Feedback control system
- Frequency/phase based - PLL
- Absolute time based – NTP, IEEE 1588

→ Oscillator tolerance (PPM)
  - Typical ±50,±100 PPM

→ 100 PPM of deviation → 100usec/sec → 8,5sec/1days
How to provide common time base?

- Message-based: actions are triggered by the reception of a message (Profibus)
- Cyclic: a periodic timing is made possible by a cyclic communication protocol (SERCOS, Powerlink)
- Time-based: system time is provided by a synchronized clock implemented in every node
  - SNTP
  - IEEE1588: high accuracy for Networked measurements and control systems
Typical application of synchronized clock

- Automation and control systems
- Measurement and automatic test systems
- Power generation, transmission and distribution systems
- Telecommunication
Automation and control systems

- Application of IEEE 1588 by different organization proposing Real Time Ethernet
  - Ethernet Powerlink (B&R)
  - EtherCAT (BECKHOFF)
  - Profinet (Siemens)

- Real time applications
  - Time stamping
  - Cyclic operation
IEEE 1588 over UDP

The Transfer Delay can be measured and eliminated. It remains an error caused by fluctuations of the transfer delay, called Jitter.

**Protocol Stack**

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

**Network**

**Delay and Jitter**
Protocol time – one step

- **Master Time**
  - $t_1$
  - sync
  - $\text{follow\_up} (t_1)$
  - $t_4$
  - $\text{delay\_req}$
  - $\text{delay\_resp} (t_4)$

- **Slave Time**
  - $t_2$
  - $t_3$
  - Timestamps known by slave
  - $t_1$, $t_2$, $t_3$, $t_4$
Protocol time – two step

Master time

Slave time

Timestamps known by slave

$t_1$, $t_2$, $t_3$, $t_4$

$sync$

$follow_{up}$

$delay_{req}$

$delay_{resp}$

$t_{-ms}$

$t_{-sm}$
PTP delay & offset

\[ \text{MS\_difference} = t_2 - t_1 = \text{offset} + \text{MS\_delay} \]
\[ \text{SM\_difference} = t_4 - t_3 = -\text{offset} + \text{SM\_delay} \]

\[ \text{MS\_delay} = \text{SM\_delay} = \text{one\_way\_delay} \]
\[ \text{Offset} = \frac{(\text{MS\_difference} - \text{SM\_difference})}{2} \]
\[ \text{one\_way\_delay} = \frac{(\text{MS\_difference} + \text{SM\_difference})}{2} \]
PTP observed drift

\[ O = \text{Offset} = \text{Clocks}_\text{Slave} - \text{Clocks}_\text{Master} \]

\[ \Delta^k = t_1^k - t_0^k \]

\[ \Delta^{k+1} = t_1^{k+1} - t_0^{k+1} \]

Drift = \[ \frac{\Delta^{k+1} - \Delta^k}{t_1^{k+1} - t_1^k} \]
IEEE 1588 Message Format

- **UDP** Port 319: event port for *Sync* and *Delay Req* messages
  
  Port 320: general port for *Follow up*, *Delay Resp* and *Mgmt* messages

- **IP** Time To Live = 0, i.e. will not be forwarded by routers
  
  Multicast addresses 224.0.1.129 for *PTP-primary* (default Domain)
  
  224.0.1.130 for *PTP-alternate1* (alternate Domain)
  
  224.0.1.131 for *PTP-alternate2* (alternate Domain)
  
  224.0.1.132 for *PTP-alternate3* (alternate Domain)

- **Ethernet**
  
  **Ethernet Frame**
  
  - Preamble
  - SFD
  - Src
  - Dst
  - L/T
  - Data
  - CRC
  
  10101011
  
  Time Stamp Point
  
  IP H UDP H PTP Message
Timestamp point

- Large unknown latency
- Small unknown latency
- Small known latency

- Only SW, Application Level
  Acc.: 100us
  Human Control

- Driver Level
  Acc.: 10ns-1us
  Process/Motion Control

- HW Level
  Acc.: <50ns
  Precision Control

Master clock

Slave clock

possible timestamp points

exchange of PTP messages
Influence of timestamp point

- Messages generate and received by PTP are delayed in an unpredictable way
  - Scheduling
  - Limited resource of CPU
  - Interrupt latency

- Constant and varying delay components
  - Transit delay
  - Jitter

- Delay effect can be bypassed with HW timestamp support
Source of delay and Jitter

Example: 100Mbps Ethernet

- Cable
  - 560 ns delay per 100m
- Hub
  - 500 ns delay, jitter about 50 ns
- Switch
  - Cut-through: minimal delay 1.12 us
  - Store-and-forward: proportional to frame length
    - 5.7..122 usec
- Router
  - Greater than Switch
IEEE 1588 Multicast Messages

Multicast Operation for Sync and Follow_up Messages

[Diagram showing the process of PTP, UDP, IP, MAC, and Phy layers for both the master clock and multiple slave clocks, with arrows indicating multicast and shared or switched medium.]
Optional improvements

- Fast synch interval at startp-up or re-synch
  - Speed-up the time of convergence
  - Non standard

- Unicast PTP Messages
  - Unicast message negotiation (optional in IEEE1588 v2 standard)
  - Limit network traffic
PTP Network Topologies

Ordinary Clock, Grandmaster: clock selected as „best Master“ (selection based on comparison of clock descriptors)

Boundary Clock, e.g. Ethernet switch

S: Port in Slave State
M: Port in Master State

Ordinary Clock
Boundary clock
Transparent clock
Transparent clock: Correction Field
IEEE 1588 Implementation

IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems

IEEE Instrumentation and Measurement Society
Sponsored by the Technical Committee on Sensor Technology (TC-9)
Software Only Implementation

Design Considerations for Software Only Implementations of the IEEE 1588 Precision Time Protocol

Kendall Correll, Nick Barendt
VXI Technology, Inc.
Cleveland, Ohio, USA

Michael Branicky
EECS Dept., Case Western Reserve University
Cleveland, Ohio, USA

Linux PTPd (source code available)
PTPd2 details

- No hardware time stamping support
- Linux kernel space timestamp
  - (SO_TIMESTAMP)

Advanced clock-servo
PTPd2 source code
Portable PTP software architecture(I)
Portable PTP software architecture (II)
PTP Implementation with IEEE1588 HW

- IEEE1588 Managed Ethernet layer2 switch
- DSP/uPC/uC with IEEE1588 MAC
- Freescale Kinetis K60 K70 and X Family
- Freescale i.MX51
- STM32
Hardware timestamp capture

[Diagram showing the flow of time between master and slave clocks, with timestamps labeled as $t_1$, $t_2$, $t_3$, $t_4$, $t_{1d}$, $t_{3d}$, and $t_{4d}$.

The diagram includes:
- Master Clock
- Network
- Slave Clock

Key points:
- PTP_RTC_SH
- PTP_RTC_SL
- PTP_RTC_NS
- PTP_RTC_NS

Timestamps:
- Sync ($t_{1estim}$)
- Follow_up ($t_{1d}$)
- Delay_Req ($t_4$)
- Delay_Resp ($t_{4d}$)

Secondary timestamps:
- SW Time Stamping
- HW Time Stamping

Counter values:
- PTP_RTC_PHASE

Time intervals:
- Every 40 ns, add 40 ns to counter

Counter types:
- 25 MHz, 5-subphase counter

Bits:
- Seconds: 32 Bits
- Nano-Sec: 32 Bits
- Sub Nano-Sec: 30 Bits

Frequency:
- 25 MHz
- 125 MHz

Output:
- Sub NS Adjustment: 30 Bits
T3LAB – PTP Porting KIT

- Based on ptpd2 source code
- Several port has been developed and validated
  - Different uC/uPC/DSP
  - Different TCP/IP stack
  - Different OS (also uC without OS)
  - Different Network Interface Driver
  - API for Ethernet micro-switch
    - With IEEE1588 hardware support (Micrel KSZ8463)
    - Without hardware support (Micrel KSZ8895)
  - Use of VLAN technologies to improve PTP synchronization accuracy
Port: TI DSP / KSZ8895

- PTPd porting for DaVinci DSP with DSPBios OS
- Network Interface improvements (timestamp)
- FPGA HW implementation of PTP clock
- Driver for KSZ8895
Port: Beagle Board/KSZ8463

- SPI HW interface for KSZ8463 eval board
- SPI driver for Linux 2.6
- Configuration API for KSZ8463
- PTP software modification (HW timestamp of KSZ8463)
Port: STR9 uC/KSZ8463

- SPI HW Interface
- PTP porting (network layer modification)
- Configuration API for KSZ8463
- PTP software modification (HW timestamp of KSZ8463)
Heterogeneous network topologies
IEEE 1588 interoperability

- EVM DM6437
- KSZ8895
- BeagleBoard
- Linux PC Browser
- Linux PC
  - PTP Master
  - Webserver
- CISCO
- MARVEL
- KSZ8463
  - Beagle Board
- KSZ8463
  - Beagle Board
Embedded switch with prioritization

- The switch maintains a separate queue per priority
- Messages containing time information can be sent with highest priority
- If highest priority traffic is moderate (e.g. exclusively caused by PTP) contention is very unlikely
- If the transmission of a low priority frame is in progress, the high priority frame has to wait until the medium is free again (worst case: 122us in a 100Mbps Ethernet)
# 802.1p VLAN Priority Mapping

<table>
<thead>
<tr>
<th>User priority (setsockopt)</th>
<th>802.1p priority (sw priority mapping)</th>
<th>Switch priority queues</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>3</td>
</tr>
</tbody>
</table>

### 802.3 Ethernet frame

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Destination Address</th>
<th>Source Address</th>
<th>TCI</th>
<th>Length etc....</th>
</tr>
</thead>
</table>

- **bits**: 16, 3, 1, 12
- **Tagged Frame Type**: 802.1p Priority, 802.1q VLAN ID
VLAN Tag/Untag example
VLAN - Sample application

- Clock synchronization via PTP IEEE1588
- Tagged VLAN traffic (802.1q)
- Measure of packet crossing time and packet loss (first-last node)
  - High and low priority VLAN Ethernet traffic
- Real-time and PTP traffic with high priority VLAN tag
- Best effort traffic with low priority VLAN tag

- 12 nodes
- MTU = 1500 Byte
- 100 Mbit/sec
- Thop = 120 μsec
- Ttot = 1440 μsec
PTP Network Demo
(Courtesy of Datalogic Automation)
Switched Network topologies

- TI DaVinci DSP with DSPBios OS
- NDK TCP/IP stack
- FPGA for PTP Clock
- Managed KSZ8895 Ethernet switch
- Daisy chain and ring topologies
- PC base application
  - PTP and VLAN validation
PTP Network Demo

- PTP Performance issues
  - CPU load
  - PTP task scheduling
  - Network traffic (vlan traffic increase performance)
  - PTP Synch message interval (1/8, 1/4, 1, 2 sec)
- **Clock-servo parameters**
  - Filter
  - PI controller
alpha = 1/2^S  
y[1] = x[0]  
y[n] = alpha * x[n-1] + (1-alpha) * y[n-1]  

Network Offset From Master (S = 1)
Clock Servo parameters offset from master filter (II)

Network Offset From Master (S=4, Ap=1000, Ai=5)
Clock Servo parameters: PI controller

```c
ptpClock->observedDrift += offsetNorm / ptpClock->servo.ai;
adj = offsetNorm / ptpClock->servo.ap + ptpClock->observedDrift;
```

Network Offset From Master (S=4, Ai =1000, Ap=10)
Clock Servo Tuning

Clock Servo Parameters Comparison

Offset From Master [ns]

S=1, Ai=1000, Ap=5
S=4, Ai=1000, Ap=5
S=4, Ai=1000, Ap=10
Giuseppe Merendino
www.t3lab.it