Introduction to the TTEthernet Synchronization Services

SAE AS-2D AS6802 Standardization Meeting, Vienna, June 2010

Wilfried Steiner
wilfried.steiner@tttech.com
TTTech Computertechnik AG
Motivation: Asynchronous / Synchronous

Asynchronous Communication

- Transmission Points in Time are not predictable
  - Transmission Latency and Jitter accumulate
  - Number of Hops has a significant impact
- Usually solved by High Wire-Speeds & Low Utilization
- Problem of “Indeterminism” remains
Given a sequence of messages on the network:

“Communication Indeterminism” means that one is not able to determine whether this sequence is correct or not, without further knowledge of the application(s) state.

- Communication Indeterminism significantly increases the complexity of system integration, diagnosis, testing, etc. (e.g. How to debug a distributed system?)
- **Synchronous** (Time-Triggered, Isochronous) Communication systematically removes Communication Indeterminism.
- Basically, two problems have to be addressed in synchronous communication:
  - Calculation of a Communication Schedule (i.e. to assign “communication slots” to nodes)
  - **Synchronization of the local clocks of the communication participants** (to be standardized as AS 6802).
Synchronous Communication

Exactly one order of messages $M_i$
(in contrast to $\text{PERM}(M_i)$ in async. comm)
Motivation: FT Synchronized Global Time

Outline:

- Synchronization Services
- Permanence Function
- Model-Based Development
Synchronization Services
Clock Synchronization Service is executed during normal operation mode to keep the local clocks synchronized to each other.

Startup/Restart Service is executed to reach an initial synchronization of the local clocks in the system.

Integration/Reintegration Service is used for components to join an already synchronized system.

Clique Detection Services are used to detect loss of synchronization and establishment of disjoint sets of synchronized components.
Clock Synchronization Service
Step 1: ALL Synchronization Master Dispatch IN Frames at the SAME Scheduled Point in Time
Step 2: Compression Master Dispatch Compressed IN Frame back to Synchronization Masters/Clients
Two-Step “Heartbeat” Clock Synchronization Service

For now, let us assume that we operate in a single-hop network:

- End Systems operate as Synchronization Masters/ Clients.
- Switches are configured as Compression Masters.

Synchronization Strategy operates in two steps for clock synchronization during normal operation mode.

**Step 1:**
Synchronization Masters send synchronization messages to Compression Masters.

**Step 2:**
Compression Masters send synchronization messages back to the Synchronization Masters and Clients.
Startup/Restart Service
Step 1: Synchronization Master Dispatches CS Frame

Compression Master

Synchronization Master 1

Synchronization Master 2

Synchronization Master 3

Synchronization Master 4

Synchronization Master 5
Step 3: Synchronization Masters
Acknowledge with CA Frame

Compression Master

Synchronization Master 1

Synchronization Master 2

Synchronization Master 3

Synchronization Master 4

Synchronization Master 5

CA
Step 4: Compression Masters “Flood” the Network with CA Frames
Four-Step Startup/Restart Service

Step 1:
Synchronization Masters send Coldstart Frames (CS) on all channels

Step 2:
Compression Masters forward all CS frames to all ports

Step 3:
Each Synchronization Master that receives a CS frame will answer with a Coldstart Acknowledgement Frame (CA) on all channels

Step 4a:
Compression Masters will forward all CA frames to all ports (for multiple failure tolerant configurations).

Step 4b:
Compression Masters will forward only one compressed CA frame (for single failure tolerant configurations).

Scenario with End System + Switch Failure
Integration/Re-Integration Service
Step 1: ALL Synchronization Master Dispatch
IN Frames at the Same Scheduled Point in Time
Step 2: Compression Master Dispatch Compressed IN Frame back to Synchronization Masters/Clients

- New device checks the number of bits set in the pcf_membership_new datafield.
- If number is equal or higher than configured threshold, than it is accepted for integration.
Step 1: ALL Synchronization Master Dispatch
IN Frames at the Same Scheduled Point in Time

IN 1: Synchronization Master 1
IN 2: Synchronization Master 2
IN 3: Synchronization Master 3
IN 4: Synchronization Master 4
IN 5: Synchronization Master 5
IN 6: New Synchronization Master 6

Compression Master

Ensuring Reliable Networks

TTTech

www.tttech.com

Copyright © TTTech Computertechnik AG. All rights reserved.
Step 2: Compression Master Dispatch Compressed
IN Frame back to Synchronization Masters/Clients

Compression Master

Synchronization Master 1
Synchronization Master 2
Synchronization Master 3
New Synchronization Master 6
Synchronization Master 4
Synchronization Master 5
Clique Detection Service
- Sender check the number of bits set in the pcf_membership_new datafield.
- If number is below configured threshold, a clique is detected.
Asynchronous Clique Detection Function

Before transmission of their Synchronization Message, the senders check the cumulative number of bits in `pcf_membership_new` datafield of Synchronization Messages received out-of-schedule.

If number is equal or above a configured threshold, a clique is detected.
Relative Clique Detection Function

- Before transmission of their Synchronization Message, the senders check the cumulative number of bits in pcf_membership_new datafield of Synchronization Messages received out-of-schedule.
- If number is equal or higher than the maximum number of bits set in the pcf_membership_new of any Synchronization Message received in-schedule, a clique is detected.
Permanence Function
Different applications have different communication needs: fault, tolerance, real time, high speed, high bandwidth, low latency, low jitter, active redundancy, hot standby, ...

TTEthernet provides following traffic classes for applications:

- Time-Triggered (TT): Bandwidth in TTEthernet networks can be highly utilized due to the possibility of strictly deterministic (vs. probabilistic) traffic scheduling of high-priority periodic traffic.
- Rate-constrained (RC): Event-triggered traffic with priority levels (ARINC 664) and transmission guarantees.
- Non-critical standard Ethernet (BE): Low priority traffic (e.g. data download) served during network idle times; can also be scheduled explicitly.

TTEthernet allows applications with different communication requirements to share a single physical network.
TTEthernet Integrated Dataflow

**Dataflow – Integration**
- Time-Triggered (TT)
- Rate-Constrained (RC)
- Standard Ethernet (BE)

Protocol Control Frames (PCFs) are transmitted on the same physical network as dataflow.
Dataflow-Integration Problem

When two (or more) messages compete for relay to the same outgoing port, the switch (or ES) has to serialize these messages.

If there are messages of same priority the messages will be serviced according First-Come First-Served.

What happens if there is a Data message (Data) in relay, when a Protocol Control Frame (PCF) becomes ready for relay?

- If a Data message is relayed by a switch when a PCF arrives, the PCF message is delayed until the relay process of the Data message is finished.
- Hence, in the worst case the PCF is delayed for a maximum sized Data message (plus other queued PCFs).
- Protocol Control Frames record their delay through the network.
- All components in the network that impose a delay on a PCF will add this delay into a field in the PCFs.
The (Dispatch, Permanence) – Pair hides the network jitter for PCFs almost entirely.
Transparent Clock, Permanence

Switch 201
- Send 302
- Receive 302
- Send 306
- Receive 306

Switch 202
- Send 302
- Receive 302
- Send 302
- Receive 302

Switch 203
- Receive 302
- Send 306

Max transmission delay (=120)
Permanence delay (120 – 10 = 110)
Permanence delay (120 – 80 = 40)

ES 102
- Dispatch 302
- Send 302
- Send 302

ES 106
- Dispatch 306
- Send 306
- Send 302

CM 1
SM 1
SM 2
SM 3
SM 4
SM 5
SM 6
SC 1
SC 2

Copyright © TT Tech Computertechnik AG. All rights reserved.
\[
\text{permanence\_delay} = \text{max\_transmission\_delay} - \text{pcf\_transparent\_clock}_n \\
\text{permanence\_pit} = \text{receive\_pit} + \text{permanence\_delay}
\]

- Frames may be delayed on their path from the producer to the consumer.
- Static delays can be accounted for offline, but dynamic delay must also be dealt with:
  - Another frame may already be in transit on a link, when a PCF frame is scheduled.
  - Relay devices may introduce dynamic delay due to non-scheduled traffic.
  - Reception may be delayed due to sequential queues and processing at the receiver.
- TTEthernet compensates dynamic delays by requiring each forwarding device to accumulate the time spent within the device in the PCF field *Transparent Clock*.
- Dynamic delays may alter the receive order of frames to differ from the send order.
- TTEthernet reestablishes the exact relative send timing of frames from different senders (besides minimal digitalization effects) called the *Frame Permanence Points in Time* (pit) by using the Transparent Clock field and assumed maximum transmission delay in the calculation at each receiver.
- Frames are delayed at the receiver until *permanence\_pit* to reestablish send time and order.
Multi-hop requires only a small add-on to the existing mechanism: an intermediate switch will not only consume the PCF, but also forward it.
Model-Based Development
Formal Verification Activities
(Synchronization Strategy)

TTEthernet Executable Formal Specification

- Using symbolic and bounded model checkers sal-smc and sal-bmc
- Focus on Interoperation of Synchronization Services (Startup, Restart, Clique Detection, Clique Resolution, abstract Clock Synchronization)

Verification of Lower-Level Synchronization Functions

- Permanence Function
  - verified with the infinite-bounded model checker sal-inf-bmc
  - using disjunctive invariant and k-induction
- Compression Function
  - verified with the infinite-bounded model checker sal-inf-bmc
  - using abstraction and 1-induction

Formal Methods have been applied as early as in the requirements capturing phase

Finalization and Completion of the formal assessment within the CoMMiCS Project

- Complexity Management for Mixed-Criticality Systems
- European Communities FP7 project [FP7/2007-2013] no. 236701
TTEthernet Executable Formal Specification

CONNECTIONS

SM1 \rightarrow \text{Frame Permanence} \rightarrow \text{Compression Master} \rightarrow \text{Frame Permanence} \rightarrow \text{Sync State Machine}

Filter \rightarrow \text{Frame Permanence} \rightarrow \text{Sync State Machine}

SM2 \rightarrow \text{Frame Permanence} \rightarrow \text{Sync State Machine}

\ldots

SMk \rightarrow \text{Frame Permanence} \rightarrow \text{Sync State Machine}

CM1 \rightarrow \text{Frame Permanence} \rightarrow \text{Sync State Machine}

CM2 \rightarrow \text{Frame Permanence} \rightarrow \text{Sync State Machine}

Current State \quad \text{Ideal State}

= or /=

DIAGNOSIS
SM, and CM describe Synchronization Masters and Compression Masters, respectively. The CONNECTIONS module describes the physical wiring plus is also used as failure-injection module.

The DIAGNOSIS module takes the current system state as input and compares it to an “ideal state”. In our case the ideal state is the one in which all correct components are synchronized to each other and the faulty components are not in a “protocol significant” state.

The DIAGNOSIS module increases a counter for each execution step in which the current system state is not the ideal state.

When we let the system start in an arbitrary system state, the maximum value of this counter represents the worst-case stabilization time of the protocol.
From SAL to Requirements to VHDL

SAL

```plaintext
[(SM.state = SM_SYNC_1070 AND
  (EXISTS (ch: TYPE_channels): message_in[ch][coldstart_ack_index].msg_type = coldstart_ack))
  -->
  SM.state' = SM_WAIT_4_CYCLE_START_CS_1050;
  SM.local_timer' = ca_offset;
  SM.local_clock' = 0;
  SM.local_integration_cycle' = 0;
  SM.local_sync_membership' = empty_membership;
  SM.local_async_membership' = empty_membership;
  SM.num_stable_cycles' = 0;
  message_out' = [[ch:TYPE_channels] empty_message];
```

Requirements:

When an end system that is in ES SYNC state receives a coldstart acknowledge frame and the end system has the es sync master flag set, it shall (a) transit to ES WAIT 4 CYCLE START CS state, (b) set local timer = es ca offset, (c) set local clock = 0, (d) set local integration cycle = 0, (e) set local membership comp = 1, and (f) set local async membership count = 0.

VHDL:

```vhdl
-- 'ReqRef{...}
if (r.state = ES_SYNC) then
  if (fpi.valid_data and (fpi.pcf_type = CONST_PCF_TYPE_CA)
    and (cdi.es_sync_master = '1')) then
    s.state := ES_WAIT_4_CYCLE_START_CS;
    s.timer := to_natural( cdi.es_ca_offset );
    s.clock := 0;
    s.integration_cycle := 0;
    s.membership_comp := 1
    s.cycles_unstable := 0;
    ...
```
Validation

System Requirements

Dataflow Requirements

Formal Model

Synchronization Strategy Requirements

Requirements-Based Testing

Model-Based Testing

Automatic Testcase Generation

Low-Level Requirements

Conceptual Design

Detailed Design

High-Integrity Requirements

Other

Time-Triggered, AFDX, Ethernet Dataflow in parallel
1 Gbit/sec
Star / Tree Network Topology
Configurable Fault Tolerance...

Requirement-Based Testing
Summary
• TTEthernet specifies synchronization services and lower-level function for fault-tolerant clock synchronization
• Formal Methods have been heavily used during the TTEthernet development
• TTEthernet Specification 1.0 is available for free download from TTA-Group Website
• TTEthernet is currently undergoing standardization at SAE as AS 6802
• Executable Formal Specification in SAL will be available shortly