Seamless reconfiguration of Time Triggered Ethernet protocols

Lukasz Wisniewski, Satendrasingh Chahar, inIT - Institute Industrial IT
Liebigstrasse 87,
32657 Lemgo, Germany
lukasz.wisniewski@hs-owl.de, satendrasingh.chahar@stud.hs-owl.de
Juergen Jasperneite
Fraunhofer IOSB-INA Application Center Industrial Automation
Langenbruch 6,
32657 Lemgo, Germany
juergen.jasperneite@iosb-ina.fraunhofer.de

Abstract

One of the key aspects to consider increasing flexibility of production plants is the industrial communication system. In case of time triggered Ethernet (TTE) systems, the flexibility is typically limited. In this paper an approach for seamless reconfiguration of TTE system is proposed. It has been shown that the reconfiguration can be performed without influencing currently running system.

1. Introduction

In order to be competitive, the future production systems have to allow fast reaction on the market turbulences, allowing to satisfy diverse customer needs. Currently, flexibility of the production systems give some angle of possible changes, however for a given price. Typically, the price is related to long retrofitting of production line that even though is possible, it is time consuming and expensive due to necessity of manual engineering and configuration of machines, robots, network devices, etc. and related to that production stops. As a consequence, there must be always a minimum amount of workpieces (the minimum number typically depends on the product complexity) that has to be produced to reach cost efficient level. The current trend of the production paradigm is the mass customisation, which should allow the customer to precisely define features of the ordered product (even at the level of the computer-aided design (CAD)). For the production system, it would mean that there might be a need for production of products with the lot size 1. Therefore, the production system should be capable to handle such requirements. The paradigm that suppose to allow offering customisation of the products keeping the production costs at a reasonable level is the agile manufacturing [1], [2], [3]. Figure 1 illustrates a fraction of an agile manufacturing system. The production line is organised in a modular way, where modules can be exchanged (e.g. Module 4 instead of Module 2) and integrated with the system promptly, according to the plug-and-produce principle [4]. Using this mechanism, after plugging in a module into a production system, it is automatically found, configured and integrated into an existing production process with minimal or optimally without any manual operations [5]. Apart from the necessity to add some new functionality, e.g. exchange of a printing module (laser instead of stamping) there are some other reasons why a module has to be added or exchanged, such as: failure of the module (or its component) or need for a maintenance. The agile manufacturing system illustrated in Figure 1 uses an advanced conveyor belt, allowing circulation of the product at the production line as long as set of required operations (e.g. screwing, painting, drying, etc.) has been performed. This approach, increases the flexibility of the production line by giving freedom in terms of modules (process steps) arrangement. This allows to keep continuity of the production process, giving a time to physically bring required modules into the system. At this production line, we can distinguish production modules that are static (always needed), e.g. some robot arm responsible for pick and place workpieces or production modules that are exchangeable, such as modules responsible to perform printing with different print quality. Each module typically consist of mechanics, electrics/electronics, control and communication system components. As shown with Module 7, each module may contain an internal network spanning devices (e.g. IO modules) that are present in this production module. Different modules may have different internal network topologies and different requirements in terms of timeliness and reliability. Apart from the internal communication, there is a need to allow communication between production modules (containing IO devices (IODs)) and between the IO Controller (IOC). Here, also some certain requirements have to be fulfiled. In order to allow realisation of such systems, there are several aspects that should be tackled at different levels of the automation system. However, the main area of considera-

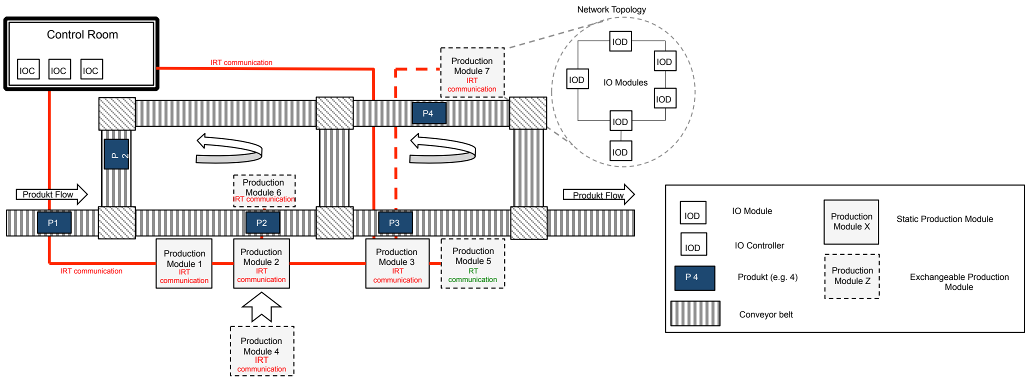

Figure 1. Agile Production Line

tion of this work is the communication system, allowing prompt adaptation to the new, not foreseen circumstances and changes and seamless integration with the currently running system configuration.

2. Time triggered communication systems

The main working principle of time triggered Ethernet (TTE) systems is that they allow communication according to a well defined plan, called communication schedule. This allows to assign critical flows to temporal time slots that are dedicated to them and can be used exclusively. Such communication systems can be used to wire the most demanding application classes, as motion control or safety critical applications. From the other hand, it provides a very good isolation for the flows that might not need such high requirements. In case of large systems, such as automotive production systems, containing thousands of network entities, it is necessary to assign them a fraction of bandwidth, in order to provide defined constant performance. This performance should be provided at any time, independent on the number of users and the current traffic load in the system.

2.1. Time Triggered Protocols Flexibility

The biggest advantage of TTE systems is the fixed high performance operation allowing to satisfy most demanding applications. The disadvantage of these systems is low flexibility that is caused by necessity of system engineering whenever communication system changes (exchange of device, new device in the setup, etc.). This is due to the mandatory definition of the network topology, setting the communication characteristics, performing the communication scheduling and distributing the communication schedule to all involved entities. This typically causes a restart of the whole system and at the same time decreases the uptime.

2.2. Reconfiguration Process

Figure 2 illustrates all timing components that can be distinguished while reconfiguring current TTE system, such as PROFINET IRT. Keep in mind that the figure presents the state of the whole system including the new device that has to be integrated.

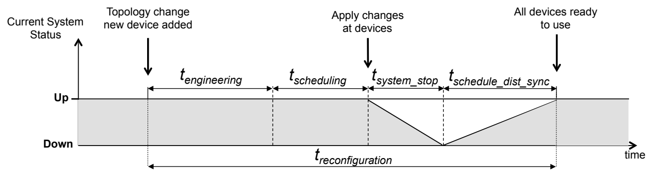

Figure 2. Reconfiguration of the system - all time components

A list of timing elements included in the $t_{reconfiguration}$ are given below:

- Engineering time $t_{engineering}$, which is the time needed to perform manual engineering of the TTE system. It is hard to estimate, how long this process could take, since it depends on any factors, such as:
  - how skilled is the engineer performing this task,
  - the fact if the correct engineering software version is installed,
  - the fact if all device description files with the right version are available.

  Since it is today still a manual process performed by an engineer, it may take between minutes up to few hours, depending on the factors listed above.

- Scheduling time $t_{scheduling}$, which is the time needed to calculate a new communication schedule. The time here depends on the complexity of the scheduling problem (topology type, constraints, no. of messages, etc.) and used algorithms.

- Before a new schedule can be distributed, the existing communication has to be stopped. This is done by sending a message from the IOC to inform IODs to enter a safe operational state. The IOD acknowledges this operation by sending a frame back. This time t_{system_stop}, depends on the network topology and its size as well as the traffic load.

- Schedule distribution time and synchronisation with the new schedule $t_{schedule\_dist\_sync}$ depends mainly on the size of the network topology and the existing background traffic. The minimal number of messages that has to be exchanged with each IOD in case of PROFINET IRT is 8 [6].

The time when the current system is down can be in the range between few milliseconds and several seconds. However, this is not the major problem. The main challenge is to bring system to the operational state as before the change. A good example is here an industrial printing machine that needs to print several copies of sheets before it does is with the required quality [7]. This generates additional costs and decreases the efficiency of the process. Therefore, integration of new devices should be performed without affecting the currently running system.

3. Proposed approach

Figure 3 represents how seamless integration of a new network component may look like. Four main time components can be distinguished here:

- Topology change detection time t_{topo_det}, which is the time needed to detect a new network device. Here, mechanisms, such as described in [5] can be used.
- Auto-configuration time t_{auto_conf}, here all needed information, such as device description, traffic characteristics, etc. will be collected. This will be used as an input for the scheduler.
- Scheduling time $t_{scheduling}$ needed to calculate new schedule. Using efficient approaches as described in [9] or [10], scheduling process can be performed even at the IOC (without involvement of any engineering tool).
- New schedule distribution time $t_{schedule\_dist\_sync}$. It is important that all devices use the same communication schedule at any time, otherwise the real-time channel cannot be used in a proper way. Therefore, this time will be defined by latest configured IOD.

In order to configure newly attached device without disturbing the currently running system, the real-time channel cannot be stopped. Therefore, the reconfiguration process has to be performed in the existing non real-time channel. In the next sections an evaluation based on the simulation model will be performed, where $t_{schedule\_dist\_sync}$ for different scenarios will be measured.

3.1. System Model

To investigate reconfiguration time of TTE system, the open source, discrete event simulation environment Omnet++ 4.4 [10] together with the INET 2.4.0 framework

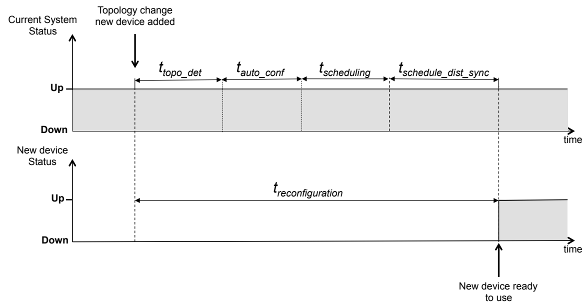

Figure 3. Reconfiguration of the current TTE system - all time components

[11] has been used.
As an exemplary TTE protocol, PROFINET IRT has been selected. This TTE protocol offers support for high variety of network topologies as mentioned in [11] and allows integration with the existing network infrastructure of the plant. The communication is organised in two phases, RED phase, were only isochronous real-time (IRT) is taking place (according to the TDMA schedule) and GREEN phase, where other kinds of traffic is allowed to be transmitted. Detailed description of PROFINET IRT can be found in [8].
To build a PROFINET IRT network device, the Medium

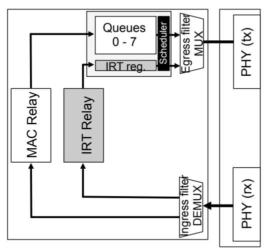

Figure 4. Hardware modification necessary to support PROFINET IRT

Access Control (MAC) layer of the INET model has been modified. The working principle is very simple, while receiving data either from higher layers or other devices, type of received data is checked and a decision is made whether this data shall be inserted into the standard tx queue of the transmission port (tx port) or it shall be inserted to the register that has been designed for IRT data, see Figure 4. Data in the register is immediately polled and transmission is performed according to the cut through mechanism. In case of a standard queue, standard Ethernet mechanism are performed, where the queues are polled according to theirs priority and forwarded according to the store-and-forward principle.

3.2 Scenarios

In this work the main interest is to investigate $t_{schedule\_dist\_sync}$ using variable cycle times and back-

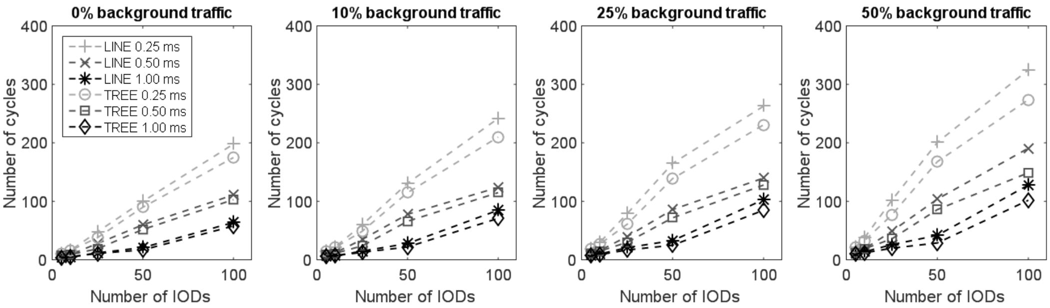

Figure 5. Reconfiguration time of PROFINET IRT network a) without background traffic b) with 10% background traffic c) with 25% background traffic and d) with 50% background traffic

ground traffic loads. Two types of topologies are considered, line and tree. The reason for this selection is that the scheduling information is distributed using best effort (BE) data channel, without using redundant connections (spanning tree only). Experiments are performed using 3 different cycle times: 0.25ms, 0.5ms and 1ms. Cycle has been equally divided to RED and GREEN phase. The background traffic will occupy 10%, 25% and 50% overall cycle time, what corresponds to 20%, 50% and 100% occupation of the GREEN phase.

3.3 Results

As expected, scenarios using line topology were more time consuming than a tree topology. It is caused by the higher distances the configuration frames have to travel, thus higher potential for queueing delays. The time increases rather linearly. In case of the worst case scenario, line with 50% load (GREEN phase fully loaded), the number of cycles needed for the reconfiguration doubled, comparing to the scenario without load. E.g. for the scenario with 0.25ms cycle time, 325 cycles were necessary to exchange 8 configuration messages (4 in both directions) for each node in the topology, what corresponds to 81.25ms. The most time consuming scenario was the line topology with 1ms cycle time, here 128ms were necessary to exchange configuration messages.

4. Conclusions and Outlook

In this work a concept for seamless reconfiguration of TTE system has been proposed. It has been measured, how many cycles are necessary in order to update devices in the topology with a new communication schedule. It has been shown that this time varies depending on the network load and topology type. If knowledge about the current background is not given, the worst case times can be taken for the activation of the new schedule. The future work will focus on investigation of even smaller cycle times (below 0.25ms) and propose optimisation of the current configuration procedure, where at least 8 messages have to be exchanged with each network device if PROFINET IRT will be used.

References

[1] Z. Zhang and H. Sharifi, “Towards Theory Building in Agile Manufacturing Strategy mdash;A Taxonomical Approach”, *IEEE Transactions on Engineering Management,*, vol. 54, no. 2, pp. 351 –370, may 2007.

[2] Y. Yusuf, M. Sarhadi, and A. Gunasekaran, “Agile manufacturing:: The drivers, concepts and attributes”, *International Journal of Production Economics*, vol. 62, no. 12, pp. 33 – 43, 1999.

[3] A. Gunasekaran, “Agile manufacturing: A framework for research and development”, *International Journal of Production Economics*, vol. 62, no. 12, pp. 87 – 105, 1999.

[4] T. Arai, Y. Aiyama, Y. Maeda, M. Sugi, and J. Ota, “Agile Assembly System by Plug and Produce”, *CIRP Annals Manufacturing Technology*, vol. 49, pp. 1 – 4, 2000.

[5] L. Duerkop, H. Trsek, J. Jasperneite, and L. Wisniewski, “Towards autoconfiguration of industrial automation systems: A case study using Profinet IO”, in *2012 IEEE 17th Conference on Emerging Technologies Factory Automation (ETFA),*, 2012, pp. 1–8.

[6] IEC, *IEC 61158-6-10, Digital data communications for measurement and control- Fieldbus for use in industrial control systems - Part 6-10: ApplicationLayer protocol specification - Type 10 elements*, 2006.

[7] H. Kipphan, *Handbook of Print Media: Technologies and Production Methods*, Handbook of Print Media. Springer, 2001.

[8] PNO, *PROFINET IRT Engineering - Guideline*, 2014.

[9] L. Wisniewski, M. Schumacher, J. Jasperneite, and S. Schriegel, “Fast and simple scheduling algorithm for PROFINET IRT networks”, in *2012 9th IEEE International Workshop on Factory Communication Systems (WFCS),*, May 2012, pp. 141–144.

[10] L. Wisniewski, J. Jasperneite, and C. Diedrich, “Effective and fast approach to schedule communication in PROFINET IRT networks”, in *2013 IEEE International Symposium on Industrial Electronics (ISIE)*, May 2013, pp. 1–6.

[11] L. Wisniewski, M. Schumacher, J. Jasperneite, and C. Diedrich, “Linear Time, Possibly Disjoint Path Search Approach for Ethernet Based Industrial Automation Networks”, in *IEEE International Conference on Emerging Technology Factory Automation*, Sep 2014, Spain.