Abstract: - This paper, as continuation of author’s previous paper, describes Ethernet Powerlink communication protocol performance evaluation. After the slight introduction to Industrial Ethernet and Ethernet Powerlink (EPL) specifically, the simulation process is described. In the following, the paper is focused on the asynchronous communication phase. The EPL cycle asynchronous phase behavior is simulated and there are shown two typical situations that can arise. In case that such a simulation is made during network design, it can reveal incorrect behavior of the network in early project phase. Economical aspects of such solution are the main contribution of such process.

Key-Words: - Ethernet Powerlink, Industrial Ethernet, Simulation, Data Link Layer, OMNet, Simulation

1 Introduction

Ethernet

Nowadays, Ethernet is the dominant networking solution for the home and office environment. It is fast, easy to install and most equipment now come with a built-in Ethernet interface. The cost of the network infrastructure is decreasing. Ethernet also allows connecting almost any type of device. Moreover, with the aid of the Internet technology, connected devices can be anywhere. With all these pros, Ethernet had a large potential to become an ideal solution also for automation technology. However, it had been known that Ethernet is not suitable for industrial networking because the medium access control method of Ethernet, CSMA/CD, exhibits unstable performance under heavy traffic and unbounded delay distribution [1].

Recently, the switched Ethernet shows a very promising prospect for real-time industrial networking, because the full-duplex access, which can eliminate frames collisions, is used. The Ethernet without collisions is no longer unstable under heavy traffic and its delays can be reduced. Therefore the switched Ethernet seems to be inherently suitable for industrial demands.

Industrial Ethernet

The adoption of Ethernet technology for industrial communication seems to be the forward step, but cannot be acceptable if any of important field area’s features should be lost. These features are namely:

- time-deterministic communication;
- time-synchronized actions between field devices like drives;
- efficient and frequent exchange of very small data records.

An essential requirement is that the office Ethernet communication capability is fully retained so that the entire communication software involved remains usable. This results in the following requirements:

- support for migration of the office Ethernet to Real-Time Ethernet;
- use of standard components: bridges, Ethernet controllers, and protocol stacks as far as possible [2].

Using Real-Time Ethernet for industrial communication brings some advantages. Well-established popular solution that uses unified physical layer leads to lower prices and vast choice of network components [3]. Link layer, compatible with the TCP/IP standard, allows addressing packets by common layer-two switches (Remark that industrial network components, which implement additional services, e.g. QoS, should be used for industrial applications). Moreover, both real-time communication and non-RT data traffic can be transferred simultaneously, although every
industrial standard solves this problem in its own way. The solution of Ethernet Powerlink, designed by B&R, is described in chapter 2.

The move from the fieldbus-based communication towards industrial Ethernet unfortunately does not mean that the industry moves from incompatible communication solutions to a scenario with all compatible Ethernet based devices. The various industrial automation protocols on top of Ethernet are in most cases not compatible with each other, e.g. a PROFINET device can not easily talk to an Ethernet-Powerlink device. Various solutions to this problem exist, e.g. through proxy devices or components or by running numerous communication stacks simultaneously, but there continues to be quite a degree of incompatibility among devices and protocols on industrial Ethernet [3]. The fact, that RTE devices can never be as cheap as devices in the office world, should not be omitted.

1.1 Practical realizations

Three classes of different approaches of RTE realization can be described. All RTE approaches have identical physical layer and thus can be connected to the universal cabling infrastructure. Non-real-time applications make use of the Ethernet protocols as defined in ISO 8802-3 and the TCP/UDP/IP protocol suite. They use typical application layer protocols like HTTP or FTP for the non-real-time applications [2].

Realization on top of TCP/IP

The TCP/UDP/IP protocol stack is used here without any modification. With this stack, it is possible to communicate over network boundaries transparently, also through routers. Hence the device almost everywhere in the internet can be reached. However, handling the protocol stack needs reasonable resources in processing power and memory and introduces nondeterministic delays in the communication. Such an approach is used by Modbus/TCP.

Realization on top of Ethernet

These realizations do not change the Ethernet communication principle, i.e. the Data Link layer of the ISO/OSI model, but are realized by specifying a special protocol type (Ethertype) in the Ethernet frame. TCP/UDP/IP protocol stack is replaced by their own protocol stack. Proper example of this class can be Ethernet Powerlink.

Realization with modified Ethernet

The Data Link Layer is modified within this approach. Most common modifications are to make the network to handle with another topology. The classical Ethernet uses the star topology. When the DLL is modified, e.g. line topology can be used without switches. The modifications are mandatory for all devices inside the real-time segment. The example of this class is the EtherCAT protocol.

2 Ethernet Powerlink

Ethernet POWERLINK (EPL) was originally developed by B&R GmbH and it is currently managed by the Ethernet POWERLINK Standardization Group (EPSP) [4]. The EPL network is based on the definition of a Data Link Layer (EPL DLL) protocol, placed on top of the IEEE 802.3 Medium Access Control (MAC) natively used by Ethernet. The EPL protocol stack is shown in Figure 1. As the picture shows, the protocol defines the application layer as well. It relies on the well known CANopen profiles. These profiles basically state that process data are to be transferred as communication objects over communication relationship.

2.1 Data Link Layer

The EPL data link layer protocol is based on a Time Division Multiple Access (TDMA) technique which grants each station with the exclusive access to the network, avoiding collisions. EPL defines two types of stations, namely Managing Node (MN) and Controlled Nodes (CNs). Each network contains exactly one MN, which represents the controller, and several (up to 240) CNs. The communication is based on the principle of using a master-slave scheduling system on a shared Ethernet segment. The interconnection between network devices may

---

**Fig. 1**
be realized either via hubs or switches. Note that hubs are actually recommended for connection, though differences between latencies and jitters on hubs and switches are negligible nowadays. Note that due to the evolution of the Ethernet technology, hubs are rapidly disappearing from the market and, consequently, their costs are increasing. The announced version of EPL, running at 1 Gbps, will make use of switches [6].

The EPL DLL protocol is based on a master-slave relationship realized by means of a continuously repeated sequence of operations as can be seen in Figure 2. The MN sends a multicast start-of-cycle (SoC) frame to signal the beginning of a cycle. Then, the isochronous period is entered. During this period, the MN sends the poll request frame PReq to each CN which, consequently responds with the poll response frame PRes. PReq frames are issued to carry output data, while PRes to carry input data. MN finishes the isochronous period by sending the PRes broadcast frame to all CNs.

Consequently, the start-of-asynchronous (SoA) frame is broadcasted by MN to notify the beginning of the asynchronous period. The SoA frame contains the specification of a CN that is granted to send one acyclic message. CNs ask for this permission during the isochronous period. When CN finishes sending the asynchronous frame, an idle phase starts. During the idle phase there is no communication on the network. This phase is intended to equate durations of adjacent cycles, i.e. to ensure broadcasting SoC frame in periodical time instants. Higher EPL layers (especially the Application layer) are not in the scope of this research and thus are not described in this paper.

3 Simulation scope definition

As the Ethernet Powerlink communication protocol is robust and well-defined, it is also very complex. For the simulation purpose it would be disserviceable to understand and express its complete functionality as it is described in specification [4]. At the very beginning stage the decision of the simulation scope should be done. Following bullets specify the scope step-by-step.

- The application layer, although contains huge amount of objects, object dictionaries, flags, control mechanisms, safety parts, is fully implemented (programmed) as a task of the Real Time OS. As a consequence of this we can say, that in case of correct implementation, the functionality of the Application layer is correct and trouble-free. The only problem is to provide powerful enough hardware; hence the application layer is not in the scope of this simulation.
- The data link layer is much more interesting. It defines the EPL Media Access Control mechanism, which is responsible for a collision-free communication between all network nodes. Figure 3 shows the time diagram of the typical EPL network consisting of one MN and several CNs.

OMNeT++ is a discrete event simulation environment. Its primarily application area is the simulation of communication networks. OMNeT++ provides component architecture for models. Components (modules) are programmed in C++, and then assembled into larger components and models using a high-level language (NED).

- simulation kernel library;
- compiler for the NED topology description language;
- OMNeT++ IDE based on Eclipse platform;
- GUI or command line environment for simulation execution;
- utilities, documentation, sample simulations etc.

4 The Model

The research is focused on standard EPL communication cycle after initialization process is fully finished. Both MN and all CNs are in their “operational” states. Firstly, both MN and CN operational states has to be analyzed. According to the specification [4], behavior has been described and transformed to the OMNeT++ programming language. Figure 3 shows illustration example of the simulated EPL network.
As we can see, there is one instance of the EPL_MN device and several (7) instances of the EPL_CN device. Apart from these devices, three Physical-Layer switches (hubs) are used for connection all components together.

Fig. 3

The main functionality of the MN is to keep the network communication. The MN starts the communication cycle in predefined intervals, serves particular CNs, starts the asynchronous phase, decides which CN is allowed to send its message during asynchronous phase. Moreover, MN processes all responses, measures time spans, guards all boundaries (e.g. cycle time exceeding) etc.

- All CNs are created as instances of one template (with different parameters). CNs are passively listening the network. In case the MN asks particular CN, this CN should process the MN’s message and send the answer to the network. Moreover, according to specific parameters, CNs can generate asynchronous events. When such an event is generated, CN tries to deliver it to the MN. Mean event delivery time is one of the most relevant network loading indicators.
- Hubs are simple switches that reproduces incoming packet to all physical ports (except of the port the packet has came) with defined propagation delay.
- Wires are metallic connection, each of them is 100 meters length and signal propagation speed is set to be similar to real metallic wires.

When simulation is run, the first network cycle starts immediately. All statistical data objects are emptied. During simulation, events are generated, transported to MN, and all important time spans are stored. When simulation is finished (according to defined end condition that can be e.g. 1 million cycles elapsed), all stored data are saved to data files and can be shown.

5 Simulation run and evaluation

As the model is implemented, appropriate parameters should be chosen. The table 1 shows relevant model parameters. The simulation results presented in this chapter describe asynchronous events occurring probability distribution on the asynchronous EPL phase utilization. For this reason, every controlled node (7 in total) obtained its asynchronous priority value. Depending on this priority, managing node makes decisions which node is served during each asynchronous phase. For the sake of simplicity, simple decision algorithm was chosen for this decision. Managing node always serves the controlled node with waiting event that has the highest priority. So called starving principle (there can come the situation when one or more controlled node will not be server at all) is used to examine network utilization.

<table>
<thead>
<tr>
<th>Network parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle Time</td>
<td>500 µs</td>
</tr>
<tr>
<td>Safety Delay</td>
<td>45 µs</td>
</tr>
<tr>
<td>Isochronous packets</td>
<td>8 µs</td>
</tr>
<tr>
<td>processing time</td>
<td></td>
</tr>
<tr>
<td>Asynchronous packets</td>
<td>24 µs</td>
</tr>
<tr>
<td>processing time</td>
<td></td>
</tr>
<tr>
<td>HUB time delay</td>
<td>1 µs</td>
</tr>
<tr>
<td>Cable length (point to</td>
<td>100 m</td>
</tr>
<tr>
<td>point)</td>
<td></td>
</tr>
<tr>
<td>Signal propagation speed</td>
<td>2·10^8 m/s</td>
</tr>
</tbody>
</table>

Table 1

Beside global network parameters, parameters specific to each module should be defined. For the desired measurement, asynchronous event priorities, asynchronous messages lengths and events probability distributions are particularly important. Table 2 shows these parameters (they are mostly identical for the sake of results readability).

<table>
<thead>
<tr>
<th>CN#</th>
<th>Async priority</th>
<th>Async frame length</th>
<th>Async distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>34B payload</td>
<td>(\text{exp}(\lambda))</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>8 B data</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>42 B total</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2

Two simulations have been performed, with different values of \(\lambda\). The network behavior was monitored. The following page shows simulation results, which are described later.
Table 3

<table>
<thead>
<tr>
<th>CN1</th>
<th>CN2</th>
<th>CN3</th>
<th>CN4</th>
<th>CN5</th>
<th>CN6</th>
</tr>
</thead>
</table>

Recent Researches in Communications and IT


156
5.1 Simulation #1
The first simulation was performed with the global parameter $\lambda = 0.02$ s. The simulation results for first 6 nodes are shown in the Figure 4. The CP (Cycle Period) mark shows the time of one full EPL cycle period. As results for all nodes are approximately the same, the network is not overloaded. We can see almost ideal serving of asynchronous events of the node 1 (the node with the highest priority). The distribution during the cycle time approximately uniform (the cycle time is just a fragment of the exponential function). Almost all events are served in the following asynchronous phase. Gradually, one node by other, the results are more fuzzy. Majority of generated events are served during the first cycle, but as the node priority is decreasing, the probability of waiting till next cycle is increasing, e.g. the probability, that there come an event in the higher priority node, is getting higher. Despite of the mentioned behavior, the network utilization is suitable.

5.1 Simulation #2
For the second simulation, the global parameter $\lambda = 0.002$ s, that means, the count of generated asynchronous events is 10 times larger than in #1. The result for first 6 nodes is shown in the Figure 5. The CP shows again the place, where the first EPL cycle ends. We can see, that with decreasing priority, the event waiting time distribution is getting wider. Events generated by first three nodes are (more or less) transferred to the MN. Many events generated of the node 4 and 5 will be probably lost. The node 6 is unable to transfer any event to the MN. We can thus see, that the network with this configuration is highly overloaded. In such a case, the modification of the network configuration must be made. Either faster network components should be used, or the network can be split into 2 independent sub-networks managed by 2 Managing nodes.

6 Conclusion
This paper presents the Ethernet Powerlink (EPL) simulation. After quick introduction to Industrial Ethernet standards and specifically EPL, the simulation scope is defined. This definition is particularly important due to high EPL complexity. When this task is done, we show on an example network the idea of simulation and evaluation of results. The simulation of asynchronous events generation probability distribution on network utilization is made and described later in the chapter 5. Two different distributions are chosen and the network behavior is analysed.

The EPL model was created as qualitative model without network timing implemented and described. Later, the measurement of real EPL network has been made. Unfortunately, this measurement could not be as precise as it should be, due to inappropriate measurement method. At present, the development of the Ethernet Powerlink Test Bed is in progress. With this Test Bed, the network timing will be measured with sufficient precision and then simulation results become more relevant. Moreover, CNs asynchronous behavior should be developed with respect to real demands (e.g. implementation of asynchronous packet flooder).

Acknowledgement
This work was supported in part by grant „Research of Modern Methods and Approaches in Automation“ from the Internal Grant Agency of Brno University of Technology (grant No. FEKT-S-11-6), and Grant Agency of the Czech Republic (102/09/H081 SYNERGY – Mobile Sensoric Systems and Network).

References: