Evaluation of the real-time capability of IEEE1394 for industrial automation

Norbert Stampfl
stampfl@ict.tuwien.ac.at
Institute of Computer Technology, Vienna University of Technology, A-1040 Vienna

Abstract - **The increasing demand for bandwidth in industrial automation requires the usage of high-speed networks in this field. Besides high data rates, the network must also guarantee deterministic timing behaviour in order to support real-time applications.**

IEEE1394, also known as FireWire™, is a relatively new high performance serial bus system, which was originally designed to connect multimedia devices. The objective of this paper is to evaluate the real-time behaviour of IEEE1394 in respect of industrial automation by analysing data transfer times and global clock support. It will be shown that IEEE1394 provides guaranteed timing constraints, and the influence of the most essential network parameters will be examined.

I. INTRODUCTION

In recent years, a lot of development was done to design high-speed wide area networks and local area networks, which satisfy the rising demands for bandwidth. But also in the field of industrial automation, the application of cameras for quality management and process supervision as well as the increasing number of intelligent sensors and actuators require a communication network with high data throughput. Up to now, industrial communication was structured in the CIM-model [1], starting with relatively simple field area networks at the bottom and developing towards more and more powerful networks on the top.

The penetration of high-speed systems to the lower layers was mainly impossible due to costs and their incapability to provide guarantee real-time behavior, which is an absolutely important criterion for automation systems at the field level. IEEE1394 supports up to 400 Mbps, isochronous and asynchronous data transfer, hot plug&play, free topology and optionally power over cable and galvanic isolation [3,4]. When considering the usage of IEEE1394 at the lower layers of the CIM-model, especially the real-time behavior is of interest. Due to its arbitration scheme, IEEE1394 generically supports real-time data traffic, even besides bulky data transmission. As a result, on one single network, nodes, supporting a real-time protocol to exchange control commands, can reside next to other legacy devices, such as a camera or storage device, while observing deterministic behavior in data transfer times. A detailed analyze can be found in chapter 3.

II. REAL-TIME COMMUNICATION SYSTEMS FOR AUTOMATION

As mentioned above, mainly field area networks are used to meet real-time communication requirements of applications for industrial automation. The simplest solution is the use of a single-master system like Interbus. As long as the application of the master is real-time capable, the whole system is deterministic. In order to achieve a more decentralized solution, multi-master systems are used with the challenging task of scheduling the access to the shared communication system in order to allow all nodes to get the chance of transmitting messages within a deterministic time interval. This can be done by using real (Profibus-FMS) or virtual token passing (P-Net). Another approach is used for CAN, where messages are prioritised. Note that only the message with the highest priority has a deterministic transfer time. Finally, TTP shall represent the systems, which provide real-time behaviour by reserving pre-defined time slots for every node. This means, that when a node does no need to transfer data during its dedicated time slot, the bandwidth is wasted. Furthermore the behaviour of the whole system has to be known a priori in order to assign appropriate time intervals to particular nodes. All these communication systems have in common, that they support data-rates up to 12 Mbps, which is fairly low in comparison to the 400 Mbps of IEEE1394. A more detailed comparison between IEEE1394 and field area networks can be found in [5,6].

As already mentioned, a lot of work is currently done to adapt Ethernet (with TCP/IP on top) to a real-time communication network. Proposals [7], dealing with traffic reduction in order to decrease the collision probability, show a lot of disadvantages, as there are the decrease of data throughput, the requirement that all nodes attached to the network support the proposed protocol and that a small, collision probability still remains.

Another approach [2] is the use of switching technology, special topologies and increasing data rates at hierarchically higher braches. Such systems require expensive switches and additional cabling costs compared with the ordinary bus topology, as every device needs a separate connection to the switch.

III. EVALUATION OF TRANSFER TIMES OF IEEE1394

As IEEE1394 covers layer 1 and 2 in respect to the 7-layered OSI communication system, the objective of this chapter is to evaluate transfer times (time interval starting at the request of a node for sending a single packet until the entire packet is really transmitted) of single packet transmissions (subactions). In order to calculate response times for application messages other parameters like packet queuing and task scheduling have to be considered and for some requests, such as read requests, two subactions one from the initiator and one from the target are necessary.

A. IEEE1394 arbitration scheme

IEEE1394 distinguishes between asynchronous and isochronous message transfer. Therefore the bus time is separated into cycles of 125µs and isochronous messages are prioritised (Figure 1). As nodes, which are intended to send isochronous data, have to allocate bandwidth and a virtual channel number at a special manager node first, dedicated bandwidth can be guaranteed. In order to save nodes, which want to send asynchronous data, from starvation, a maximum of 80% of the total bandwidth can be allocated for isochronous data traffic.

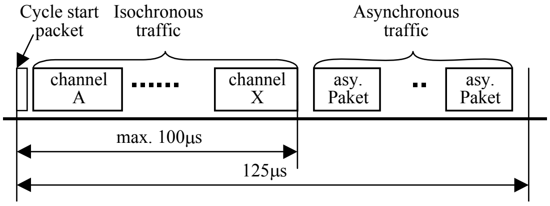

Figure 1. IEEE 1394 isochronous cycle

The subsequent calculations will show how several transfer times depend on the number of hops, the number of nodes and the packet size. As the maximum packet size is proportional to the data rate of IEEE1394, the packet size is normalized (speed={1,2,4}), e.g. for 100 Mbps the maximum packet size is 128 quadlets (512 bytes). Note, that the packet contains payload as well as a header of 3-8 quadlets, depending on the packet type.

Furthermore it is assumed, that the IEEE1394 parameter gap_count is optimised and standard cables are used, in order to represent the influence of the topology by the maximum number of serial hops (h≤23). As IEEE1394 nodes are connected peer-to-peer, a hop represents one single connection between two nodes.

Nodes, attached to an IEEE1394 network, are able to determine the total number of nodes and hops as well as the maximum packet size each node is able to transmit. Every node can therefore verify if the currently available network is able to fulfill its timing constraints.

B. Isochronous transactions

For a node, which has allocated bandwidth and one or several channels (up to 63 per network), isochronous packet transmission is guaranteed every 125µs per channel on an average. As for real-time systems, the transfer jitter is an important parameter, the minimum and worst-case maximum time a node has to wait until one entire isochronous packet is transmitted needs to be evaluated.

Minimum transfer time

In order to calculate the minimum transfer time, we assume, that the bus is idle. Since we do not consider queuing or handling times neither in sending nor in the receiving node, the sender can immediately start to arbitrate and transmit the packet on the bus. According to the IEEE1394 standard [3,4], an isochronous subaction consists of arbitration, packet transmission and the isochronous gap (1).

$$t_{iso\_transfer\_\min} = t_{arbitration} + t_{data} + t_{iso\_gap} \qquad (1)$$

The most important parameters are the length of the packet, the used speed and the maximum number of hops.

Figure 2 shows, how the transfer time depends on the number of hops and the packet size.

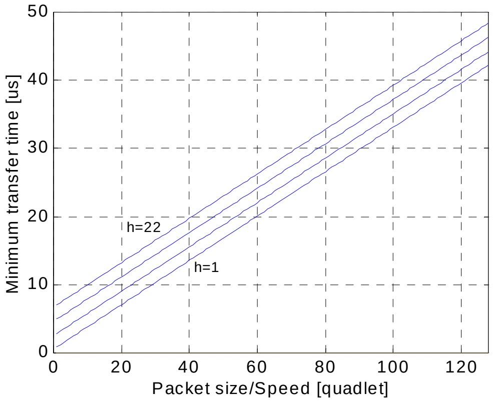

Figure 2. Minimum isochronous transfer time

Maximum transfer time

Since allocated bandwidth does not need to be used (and remains for asynchronous transactions), asynchronous traffic might immediately follow the cycle start packet. If a node requests a bus for transmission of an isochronous packet (e.g. on channel C in figure 3), it has to wait for the subsequent cycle. The worst case occurs, when an asynchronous packet postpones the cycle start packet, and other nodes make use of their allocated isochronous bandwidth first.

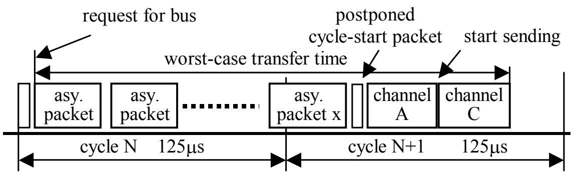

Figure 3. Worst-case delay for a single isochronous packet transmission

The maximum time, a node has to wait for the transmission of isochronous data on a particular channel can be calculated by

$$t_{iso\_transfer\_\max} = (1+0.8)\times t_{cycle} + t_{asy\_\min} = 287\mu s \quad (2)$$

It is dependent on the amount of allocated isochronous bandwidth and the maximum asynchronous packet size used by applications on the bus. Note, that this is the worst case for one single packet. If the node has collected more packets, it is able to send one packet per isochronous channel every 125 µs on an average. Obviously, the transfer jitter is relatively high. For applications, requiring smaller jitter, time stamping by means of the global clock is proposed. The receiver does not acknowledge isochronous packets. This means, that additional overhead might be necessary for higher protocol layers to ensure error detection and handling.

C. Asynchronous transactions

Obviously, isochronous transactions support guaranteed transmission times, but due to the fairness interval mechanism, asynchronous packets are also transmitted within deterministic time limits. The mechanism is based on the assumption that a node is supposed to send only one packet per fairness interval. If all nodes are blocked because the have already sent a packet or do not want to arbitrate, the bus is idle for some time and the fairness interval is restarted. Therefore the duration of the fairness interval varies, resulting in significant jitter for the maximum transfer time. The fairness interval and the isochronous cycle are independently interlaced.

Minimum transfer time

Similar to isochronous subactions, an asynchronous subaction can immediately take place if the bus is idle. An asynchronous transaction consists of different phases: arbitration, data and acknowledgement and the subaction gap. Therefore the minimum transfer time is given by

$$t_{asy\_transfer\_\min} = t_{arb} + t_{data} + t_{ack} + t_{subaction\_gap} \quad (3)$$

Figure 4 shows the influence of the speed, the data-size and the number of hops on the minimum transfer time.

Maximum transfer time

Assume, that node N1 has the lowest priority, but sends a packet at the beginning of a fairness interval, because no other node requests the bus. If during the transmission of the packet, the other nodes collect data, N1 has to wait two fairness intervals in the worst case to transmit its subsequent packet. The longest duration of this interval can be observed, when all nodes (≤63) send packets with the maximum amount of data. Consider, that this is a very rare case and that the average transfer time will be much shorter. In Figure 5, it is assumed, that all nodes are daisy chained up to the maximum number of hops (worst-case topology) and that every node wants to send a packet.

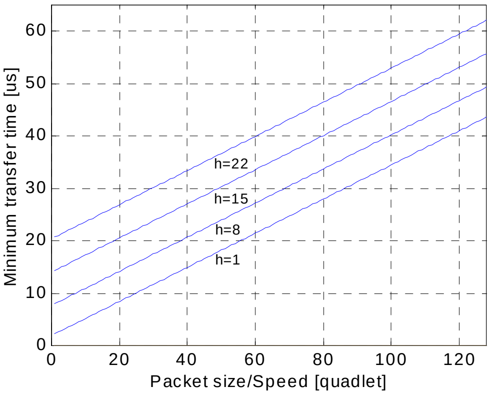

Figure 4. Minimum asynchronous transfer time

For the transfer time, the average size of the packets is essential. As real-time applications usually transfer data packets of small size, the guaranteed worst-case transfer time should not exceed 2 ms. Consider, that if a node has collected more packets for transmission, it is allowed to send one packet every fairness interval.

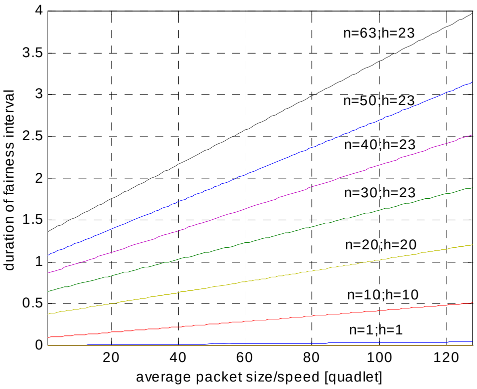

Figure 5. Maximum duration of the fairness interval

In case of the usage of both isochronous and asynchronous traffic on the same network, the transfer times for isochronous messages are not concerned. For asynchronous subactions however, depending on the amount of allocated isochronous bandwidth ($B_{allocated}$), the maximum transfer time is extended up to

$$t_{asy\_transfer\_\max\_iso} \approx \frac{t_{asy\_transfer\_\max}}{1 - 0.8 \times \frac{B_{allocated}}{B_{allocatable}}} \quad (4)$$

If all isochronous bandwidth is allocated and used, the maximum transfer time is approximately 5 times higher than without isochronous traffic.

D. Impact of bus resets

Bus resets are very critical events concerning the real-time behaviour of an IEEE1394 network. They are forced by any change in the topology, e.g. when nodes are removed or attached to the bus. After the occurrence of a bus reset, all data traffic is interrupted and the bus is reinitialised, which can take up to 250 µs. Although isochronous traffic can continue immediately afterwards, the nodes have to explore the network for their communication partners, because the node IDs, used for addressing of asynchronous packets, may have changed. This means, that the asynchronous application data traffic is blocked during a significant and possibly non-deterministic time interval.

For real-time applications the occurrence of IEEE1394 bus resets is therefore not allowed. As a consequence all calculations above assume, that no changes in bus topology occur. For networks in industrial applications, this requirement is easy to fulfil, as they are usually installed once and modified only for maintenance.

IV. GLOBAL CLOCK SUPPORT

As data in real-time systems is only valid for a particular period of time, a synchronized global clock is absolutely vital. Furthermore, real-time applications often depend on a constant transmission delay of data received from other nodes, requiring minimal jitter in packet transmission of the communication system. Since this cannot be guaranteed generically by IEEE1394, the time stamping of time critical data is proposed, which also requires a global system clock.

IEEE1394 nodes maintain an internal clock with 40.7 ns ticks. If there is a cycle master available on the bus, being mandatory for isochronous traffic, these internal clocks are updated every 125 µs by the cycle start packet. Assuming that all nodes update their clocks in the same way, which is usually done by hardware, the jitter of the clocks only depends on the propagation delay of the cycle start packet. In the best case all nodes are directly connected to the cycle master. In the worst-case, the cycle master is at the end of a daisy-chained network. If we assume a typical propagation delay of 144 ns per node [3] and a typical cable delay of 227 ns (4.5 m standard cable), the variation of an individual clock from the clock of the cycle master can be determined by (5).

If less clock variation is desired, the applications can adjust the internal clock by a constant time delay (nodes can determine the number of hops to the cycle master) or use additional clock synchronization mechanisms.

$$t_{clock\_variation} = 371\,ns \times h_{hops\_to\_cycle\_master} \tag{5}$$

V. CONCLUSION

In order to determine, whether IEEE1394 is suitable as a communication system for real-time applications or not, minimum and maximum transfer times of isochronous and asynchronous subactions have been calculated, as well as global clock behavior. IEEE1394 supports deterministic and guaranteed maximum transfer times as long as bus resets, respectively changes in bus topology are avoided. The timing significantly depends on several bus parameters, such as speed, the number of hops, the number of nodes and the packet size. As all these parameters can be extracted during operation, the applications or a simulation of a particular network configuration can determine a priori if the required real-time behavior is achieved. Due to its arbitration scheme, the packet transfer jitter is almost equal to the maximum transfer time, which could be unacceptable for several applications. To compensate the jitter, time stamping of application data by the use of the global clock proved to be an appropriate solution. The big advantages of IEEE1394 in comparison with other real-time communication systems are its high data rate, the efficient use of bandwidth and that worst-case transfer times are also guaranteed if other protocols like TCP/IP or SBP-2 are used on the same network.

REFERENCES

[1] G. Schickhuber, O.McCarthy: Distributed fieldbus and control network systems, C&C Journal 1997

[2] S. Rueping et al.: Analyses of Switched Ethernet with Different Topologies Used in Automation Systems, Proceedings of FeT'99, Springer Verlag 1999

[3] IEEE1394-1995; Standard for a High Performance Serial Bus, IEEE 1995

[4] P1394a5.0; Draft Standard for a High Performance Serial Bus (Amendment), 1394 Trade Organization 1999

[5] L. Ruiz, Ph. Dallemagne, J.D. Decotignie: Using FireWire as Industrial Network, IEEE1999

[6] N. Stampfl: IEEE1394 in Comparison with Other Bus Systems, Proceedings of FeT'99, Springer Verlag 1999

[7] F.Furrer: Ethernet TCP/IP für die Industrieautomation, Huethig Verlag 1998