Ethernet TCP/IP in Automation
A short Introduction to real-time Requirements

Max Felser
Bern Institute of Engineering and Architecture
Bern University of Applied Sciences
Morgartenstrasse 2c, CH-3014 Bern
SWITZERLAND

Abstract **- Since more than a decade the experts try to convince the users, that Ethernet is not suitable for applications in real-time communication. Only fieldbus are well adapted to the real-time problems in the automation applications. Today several manufactures propose networking solutions based on the Ethernet-TCP/IP networks. This paper presents examples of problems to be solved, when a user intents to replace fieldbus networks by Ethernet-TCP/IP based networks for automation applications.**

I. INTRODUCTION

Over several decades the fieldbus adapted to the needs of automation environment: Easy manipulation by installation personal, robust in dirty environment, cheap and simple cabling. The star-based traditional cabling is replaced by the more efficient bus- or tree-topology. Also on the technical point the fieldbus improved the communication models and technology to the needs of real-time control information.

On the other hand we see that Ethernet and TCP/IP is widely accepted in the Information Technologies (IT) and allows simple access and integration to the world-wide internet networks and technology. In clean office environment Ethernet and TCP/IP are the standard communication network and therefor very cheap and well known by a lot of IT-Experts. If we want to use this technology also for automation applications, we have to adapt the physical interfaces and installation technology to the "field" environment. The transmission of real-time control Information is demanding also some special requirements. This paper focus on these real-time requirements and lists possible solutions to fulfil these requirements.

II. DEFINITIONS

There are different requirements which can be defined for real-time communication: Latency, receive variance and scan time jitter. The throughput is more important for office applications and is not further discussed here.

A. Latency

In a simplified model a sender wants to signal an event by sending a message to a receiver. The maximal delay of this message - the latency - is a possible real-time constraint, to be fulfilled by the communication system. The latency is composed of different part delays with different reasons.

1) Preparation of message: First the sender has to prepare the message and run though the upper layers of the protocol stack. This takes the time Δt_1.

2) Wait for access: Once the message is ready to transmit, the sender has to wait until the medium access control (MAC) gets control over the communication system. After Δt_2 the message gets transmitted.

3) Transmission: The transmission of the information over the network takes Δt_3 time delay.

4) Message received: The message is processed by the receiver and passed to the destination application. For this an additional delay has to be considered.

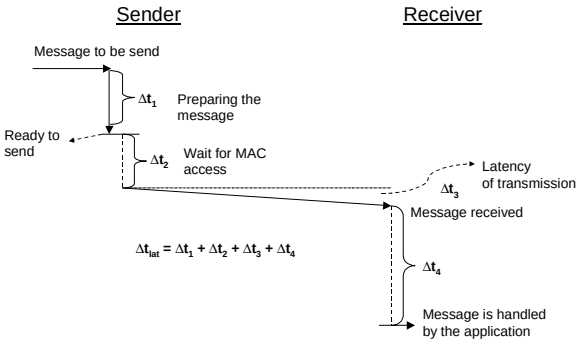

Fig.1 Latency of a message

The latency in this simple model is the sum of all part delays.

B. Receive variance

Very often in control applications the same value must be used on two different places at the same time. For consistency reasons the value should be the same in a certain time delay. This receive variance Δt_{rv} depends on the requirements of the application.

There exist two variants: The system sends one message for every destination or there are multicast messages used.

In the of unicast messages, the latency of the message adds together, to form the receive variance.

In the multicast message, only the differences of the delay of the network equipment leads to a receive variance.

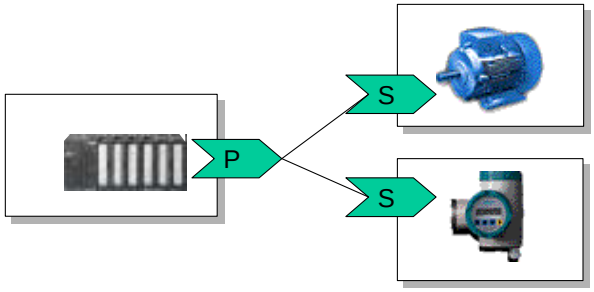

Fig. 2 Receive variance if the latency of one message to different receivers is different

C. Scan time jitter

For distributed control-loops the messages have to be transmitted in cyclic intervals to all devices involved in the loop. That the control-loop can work correctly, the scan time has to be implemented with a limited jitter. In practice, a jitter of than 5% of the scan time is requested.

The scan time varies on the sum of the receive variance of the different messages.

The effect of the jitter can be limited, if a multicast message is used to synchronise the field devices, as shown in fig. 3. In this case the jitter is limited to the receive variance of the multicast message. This holds as long as the scan time is not longer as the cycle time of the application.

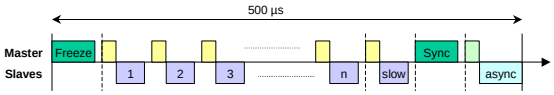

Fig. 3 Scan cycle with synchronisation message

III. TECHNICAL SOLUTIONS

We want now compare, how fieldbus and Ethernet TCP/IP communication systems fulfil the requirements regarding determinism, queues, consistence and cyclic control.

A. Determinism

The request is, that the latency for a message Δt_{lat} never takes longer than an application specific maximal delay. Most part-time of the latency can be calculated. The most critical part is Δt_2.

Fieldbus are build around this constraint. The simplest solution is to build a master-slave system, where the master controls the bus access and the possible maximal delay can be calculated based on the polling algorithm of the master.

Ethernet permits collisions and has therefore no deterministic behaviour of the medium access.

There exist three possibilities to get a more deterministic behaviour of the Ethernet:

1) Limited traffic: Reducing the traffic on the network to reduce the number of collisions is the simplest solution. With a reduction to less than 10 % the risk of non deterministic and unacceptable delays is about null.

2) Master-Slave: The easiest solution is to put a master-slave system on top of Ethernet. This will give a deterministic system similar the one used by fieldbus.

3) Switched: The most proposed solution today is to avoid the collisions by segmenting the bus with a switch in several "Collision Domains". In best case you get one domain for each device and no collisions anymore.

For the *limited traffic* solution you separate the real-time network from the IT world by routers or bridges. You have to ensure that the network is oversized in network usage. This is a non-efficient solution. It is used and proposed already for tens of years by different manufacturers.

In the *Master-Slave* solution one device is the Master and polls the other as slaves. Normal IT devices, not designed for these Master-Slave rules, are not allowed to be on the same segment. This solution is most efficient, when no switches, only Hubs are used and is proposed in [1].

The *Switched* solution has the major inconvenience that switches are blocking or – at least at the moment – still very expensive devices.

In all solution with high bitrates we introduce a star-topology cabling system, we wanted to avoid with a bus-topology network.

B. Queues

To reduce the MAC time Δt_2 and the transmission time Δt_3 it is common practice to increase the bitrate speed of the communication system. More messages are coming now to the receiver in the same time. If the receiver is not able to handle all these messages at the speed they get in, they are queued in the communication stack (Fig.4).

The same holds for switches: if the communication capacity between the switches is lower than the sum of the capacity to the end users, the switch has to drop or queue the messages. This adds an additional delay.

To get e.g. the actual value of a sensor, the queue has to get emptied first. In a real-time control system this means, that the delay Δt_4 is increasing, if the receiver or the trunk is not able to handle the speed of the network.

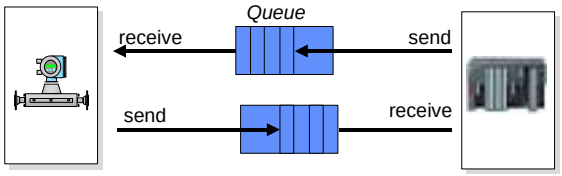

Fig. 4 Message queues in the communication system

Fieldbus do not work with queues for real-time control information. They use a buffer mechanism. The new message overwrites the old value in the buffer (Fig.5). This allows efficient solutions even with low computing resources. This buffered communication works with cyclic or event-driven communication. This model is based on the idea, that only the last valid value is interesting and not the sequence of values.

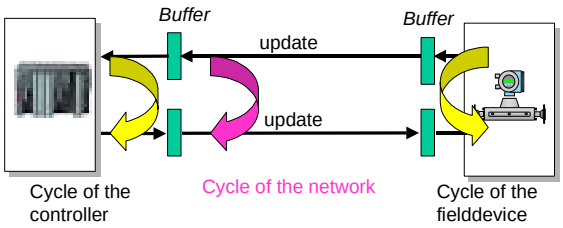

Fig. 5 Buffers for cyclic real-time control information

For the adaptation to Ethernet and TCP/IP there exist different solutions to this problem:

1) Priorities: Define a clear strategy for IP priorities and define separate queues for real-time control information.

2) New: Invent a new protocol on top of Ethernet based on the buffered principles or use an existing fieldbus protocol on top of Ethernet replacing TCP/IP.

The priority solution has the advantage, that with the implicit usage of priorities in a automation systems design, there is no need to modify the communication stack. However UDP telegrams will be used and not TCP!

In the case of an invention of a new protocol, there is no compatibility with the existing IP Protocol anymore.

C. Consistence

IT-communication systems work with the Client-Server Model. This communication model is very useful if there are point-to-point communication relations with a clear hierarchical relation. The Client requests the Server to provide a service on an communication object (fig.6).

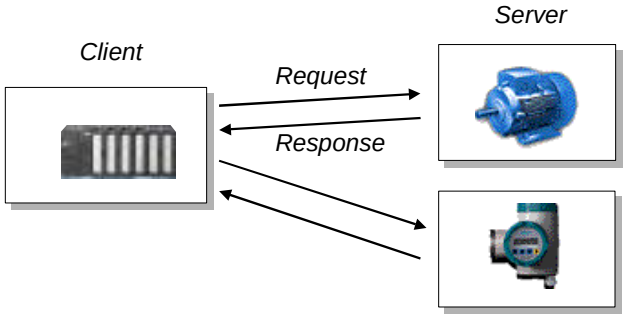

Fig. 6 The Client requests a service from the server

This communication model was also defined for automation systems. The Manufacturing Message Specification (MMS) was used as a model for most fieldbus specifications. It works well, if you have to control a field device or modify parameters in the field device. But there are problems to implement consistent communication in a distributed system with the Client-Server model: If you want to set a value in two field-devices from a controller in a consistent way, you have to define two Client-Server relations. On one communication network, these service request messages are always passed one after the other, so there is automatically a $\Delta t_{cons} > \Delta t_{lat}$.

Distributed control-systems based on fieldbus use a different approach: the Publisher-Subscriber Model.

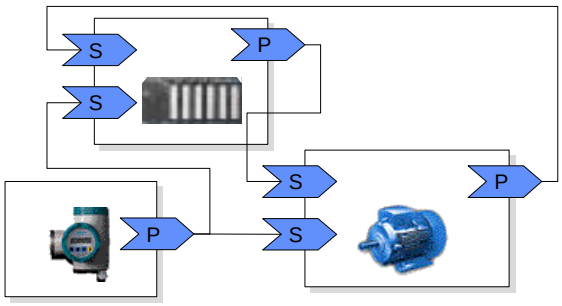

Fig. 7 Published (P) values are taken by the Subscriber (S)

The consumer of a value subscribes to the publisher of an output variable of a function. The producer of the variable publish the actual value on the communication network and all subscribers can take the same value on the same time from the same message. The only difference in delay is the difference in transmission delay Δt_3 due to the signal-runtime on the communication cable and the different Δt_4 in the receivers.

To get acceptable consistence in Ethernet based network there exist different approaches:

1) speed: Just increase the bitrate on the system. This reduces mainly Δt_2 and Δt_3 and reduces therefore also $\Delta t_{lat} < \Delta t_{cons}$.

2) Fieldbus: Use an existing fieldbus protocol on top of Ethernet which implements the Publisher-Subscriber model.

3) New: Invent a new protocol on top of Ethernet based on the Publisher-Subscriber model.

The first solution to increase the bitrate creates the problems with the queues described in the previous section. The only solution there was to invent a new protocol, which comes to the same as the other solutions in this section.

D. Cyclic control

For distributed control-loops the messages have to be transmitted in cyclic intervals with a limited jitter.

Fieldbus use for this a central controller: the scheduler. The central scheduler ensures, that the timing constraints on the bus system are fulfilled and sends the corresponding synchronisation messages. In a polled system it calls all values in the correct cycle interval.

For Ethernet different solutions are possible:

4) Master Scheduler: Use the fieldbus master-slave principle also over Ethernet and get the cycle control like this.

5) Time Stamps: Synchronise the clocks in all devices. The messages are send with time stamps. The receiver corrects the different receive variant to the specified max. latency.

The Master Scheduler has the disadvantage that the availability of the system depends on the availability of the central scheduler.

With the second solution it gets possible to get cyclic communication with limited jitter, even with communication which is not very stable. However this needs additional computing resources for the correction of the delays and local clocks which are sufficient exact for the needed time stamps.

IV. CONCLUSION AND OUTLOOK

Most problems for real-time control information over Ethernet TCP/IP are identified. Different approaches listed are realised or are under development. It is not possible to fulfil the requirements of real-time communication over Ethernet with the existing TCP/IP protocol. It will be replaced by the Priorities and UDP/IP. To get the application models of distributed control working, a new Middleware is needed. This Middleware will implement the Publisher-Subscriber Model or another fieldbus oriented protocol including the priority mechanism and time synchronisation.

There exist private solutions of one manufacturer or open solutions which are developed by different manufacturers together for these Middleware already. In this session different proposed solution of IDA, IAONA and PROFINet are presented.

V. REFERENCES

[1] A.Meindl, " Ethernet, 'one slot at a time,' becomes a synchronous network, *Control Engineering Europe*, Cahners Europe, June/July 2001, p.20