IEEE 1588 Real-Time Networks with Hybrid Master Group Enhancements

Georg Gaderer
Institute of Computer Technology
Vienna University of Technology
Gusshausstrasse 27-29
A-1040 Vienna, Austria
Georg.Gaderer@TUWien.ac.at

Patrick Loschmidt, Thilo Sauter
Research Unit for Integrated Sensor Systems
Austrian Academy of Sciences
Viktor Kaplan Strasse 2
A-2700 Wiener Neustadt, Austria
{Patrick.Loschmidt, Thilo.Sauter}@OEAW.ac.at

Abstract

The widespread clock synchronization standard, IEEE 1588, is purely master-slave based. The inherent disadvantage is, that a failure of the master requires the election of a new one, during which the network nodes cannot be synchronized. The present paper proposes a compatible extension to the standard introducing an architecture for fault-tolerant and seamless distribution of time information.

After the introduction of the architecture the common application of Ethernet is treated to outline the problems arising from this network type. The proposed solution significantly increases the accuracy of the synchronized network in case of failures and additionally provides backup strategies in case of network problems.

1 Introduction

It is known that engineering communication in a distributed system can benefit from synchronized clocks on every node. One very common application for synchronized clocks are network protocols which rely on the usage of time-slot-based medium access control like TDMA. In that particular architecture a time slot is assigned to every network participant, in which the node – and only the node – is allowed to send its data. Therefore data transmission can be guaranteed and the network itself can thus be considered as real-time network [1]. Another strategy is the synchronization on the upper layers: If the application can be structured in a time triggered fashion, any pending operation gets scheduled by the application (in spite of the communication system) and a hard real-time execution of actions in the distributed system can be guaranteed. Nevertheless, the network has to provide a given bandwidth in those cases, so that not only the message execution but also message delivery can be guaranteed.

Other applications would be logging and data collecting tasks where not the delivery time, but the sample time of data is an essential benefit from synchronized nodes. Examples for such applications are fault detection in energy supply lines or volume balancing in water distribution grids. The latter helps water suppliers to detect leakage and theft: By making snapshots of water meters at predefined times a balance can be drawn, and by comparing it with the amount of water feed-in manipulation of the supply network can be detected. Last not least, apart from the TDMA-class protocols and the synchronized applications, clock synchronization is often also needed for the communication itself: For example, some powerline networks use a master-slave based communication protocol. In that case the master generates a frame clock which is evaluated by the slave to identify a start of frame and slots for alarm messages. In environments like powerline communication, links can be lost at any time due to the heavy distortion on the medium. Even worse: since it is common in energy networks to do load balancing by switching whole subnets from one transformer station to another, also all master-frame clocks have to be synchronized. Further more, it is reasonable to avoid the time-consuming re-logon process. Therefore a highly precise timebase must be kept on the node to ensure an accurate time even if no reference clock signal can be provided by the powerline master.

However, addressing the above mentioned application fields and considering the properties of time as a communication variable, fault tolerance has to be included in this kind of real-time networks. To ensure a highly stable reference time, the clock values are distributed from reference nodes to clients. If, due to architectural reasons, all reference nodes are shrunk to one single clock node in the system, this node is a typical single point of failure. Even if a second reference node is provided on a hot standby basis, the system is out of sync during the time-consuming switchover to the new clock reference. The aim of this paper is to enhance master-slave fashioned IEEE 1588 networks in a way that not a designated master is used as reference time node, but to include *all* nodes with high precision. This allows to get rid of the master which is a single point of failure. The proposed methodology causes the system to run with reduced accuracy in the worst case but keeps the attached nodes synchronized under all circumstances.

The remainder of this paper is structured as follows: Section 2 gives an overview of the state of the art. Section 3 analyses the master failure problems for the specific case of Ethernet and addresses also additional fault tolerance issues of the same. Finally, Section 4 comes up with the proposed enhancements which are also discussed in the conclusion.

2 State of the Art

Although clock synchronization is well investigated in the academic laboratory environment, recent developments in factory automation – especially the idea of using standard-office Ethernet on the factory floor – revived the topic. There are essentially two competing paradigms in clock synchronization: the democratic and the master-slave style approaches.

2.1 IEEE 1588

The story of a simple clock master-slave synchronization with adequate performance in real-time networks is a quite successful one. Since the publication of the IEEE 1588 many standard industrial products adopted the standard [2, 3]. The concept of IEEE 1588 is as simple as effective: After power-up a so-called master election is initiated to elect the most appropriate network node to provide reference time. Clock synchronization is then done subsequently in two steps. First, the delay between master and slave is measured via so-called delay-request and delay-response messages. This step is needed to enable the slaves to eliminate any transport delay between the two protocol stacks. With this delay information the master sends a sync packet, succeeded by a follow-up packet. This follow-up packet delivers the exact send-time of the previous packet, allowing the slave to cancel out the now known transport delay. Note that IEEE 1588 is a protocol definition and not limited to a specific physical network implementation.

2.2 Democratic Approaches

Besides the fully master-slave oriented clock synchronization also democratic algorithms are well investigated [4, 5, 6]. These algorithms take the time of all reference clocks and combine the values to one ensemble clock value. A very basic approach would be to combine all samples by calculating the mean value. This solution has the obvious disadvantage that any byzantine (or babbling idiot) node has the potential ability to decrease the ensemble accuracy. The same is true for the simple case that some clocks might be wrongly adjusted. Advanced algorithms like the external clock synchronization [7] ensure that at least $2F + 1$ reference time servers are needed to mask F arbitrary failures of reference time servers. The same requirements are addressed by [8]. The difference is that locally not only the clock value is maintained rather a confidence interval. This interval is adjusted according to the local precision. This has the advantage, that nodes which are included in the ensemble time can be weighted according to their internal structure. E. g., a GPS clock delivers a highly precise externally aligned clock if the transceiver has connection to his satellites, which is not true if the link to the satellites fails. Yet an OCXO-controlled non-GPS node is, once it is synchronized, able to maintain a high accuracy clock.

3 Problem Definition for Ethernet

The most obvious single point of failure of this protocol is the master-slave principle. When a master fails (e. g., no response to delay-request messages, or the stopping of the regularly transmitted sync messages) a new master election is initiated. Following the IEEE 1588 standard, this reelection of a master takes place after 10 sync periods elapsed without message. During that time all slaves in the system are running freely, i. e., the accuracy is determined only by the accuracy of the local oscillators, which is in turn influenced by many external parameters like temperature, age, and electrical load. Thus, given the IEEE 1588 time range of a sync-interval specified as $\{2^0\,\mathrm{s} \ldots 2^6\,\mathrm{s}\}$, the absolute error for a node with a COTS oscillator with 100 ppm stability can be calculated as $10 \times \{2^0\,\mathrm{s} \ldots 2^6\,\mathrm{s}\} \times 100\,\mathrm{ppm} = \{1\,\mathrm{ms} \ldots 64\,\mathrm{ms}\}$, which seems too imprecise even for systems with a high synchronization rate.

A second issue focuses on the case of Ethernet, where the obvious single point of failure is the Ethernet switch. Therefore any possible solution has to address the problem that a failure of a switch must neither cause the communication protocol to stop synchronizing the nodes, if they are still reachable, nor significantly reduce the overall accuracy. A completely democratic architecture also does not seem to be reasonable, since in a network with n participants every node has to tell the remaining $(n-1)$ peers its local time together with the confidence interval. All other nodes need to distribute that time, too, and therefore

$$M_{\mathrm{democ}} = n \times (n-1) \tag{1}$$

unidirectional communication links have to be established to distribute the synchronization data, where M_{democ} is the number of links. This is significantly more than the $(n-1)$ links required for the strict master-slave principle of IEEE 1588. Furthermore, the efficiency η decreases with the number of nodes,

$$\eta_{\mathrm{democ}} = \frac{n-1}{n \times (n-1)} = \frac{1}{n}. \tag{2}$$

4 Proposed Solution

4.1 Master Groups

The considerations of the previous section lead to a three-level architecture consisting of hierarchically structured synchronization subnets (SSNs) [9]. The reference time is broadcasted by GPS satellites and atomic clocks, respectively. The GPS receivers, which are considered as reference clocks, are coupled directly to the nodes of the master group which can be interpreted as a fully democratic subnet where each member of the group talks to all others.

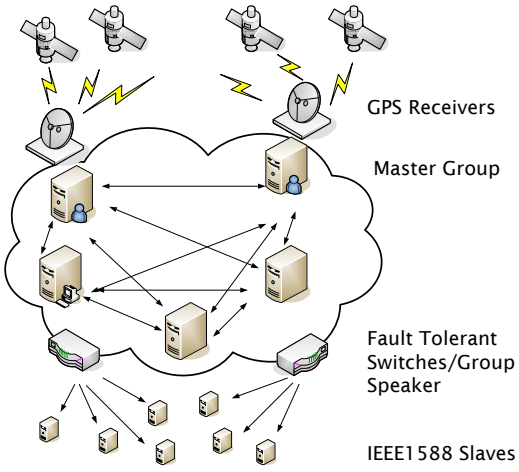

Figure 1. Master group concept

This approach has the advantage that a failure of a single master has hardly any influence on the slaves associated, except that the overall accuracy of the master group is reduced. Within the master group the democratic SynUTC [10] protocol is used to synchronize the participating nodes. Nevertheless, other democratic and fault tolerant approaches can be used as well. [11, 12]

The mastergroup nodes determine a fault-tolerant average value of the current time and pass it on to all IEEE 1588 slaves. The transmission takes place via a so-called master group speaker, represented by the switches in Figure 1. This speaker and the switches for the cross-linking of the master group must also offer the possibility for redundancy on the physical and on the protocol layer.

The group speaker communicates with a set of standard IEEE 1588 (version 2000) slaves, which ensures low traffic volume (compared to the reference clock interconnection) even for high numbers of slaves. The associated master for each node is the speaker of the superordinate group which acts transparently like an IEEE 1588 master and passes the ensemble time from the group downward. The very heart of this approach is to enhance IEEE 1588 networks with this transparently integrable master group to a hybrid architecture in order to increase stability and fault tolerance.

The efficiency for m masters (including the group speaker) with n nodes compared to the IEEE 1588 master-slave principle

$$\eta_{\text{hybrid}} = \frac{(m-1)+n}{m \times (m-1)+n} = \frac{m+n-1}{m^2-m+n}. \quad (3)$$

Note that for the common case of only a few masters synchronizing substantially larger number of client nodes ($m \ll n$) the complexity approaches the one of master-slave method.

4.2 Fault-Tolerant Switches

A further central step are master group- and IEEE 1588 time-aware switches offering the possibility for fault tolerance. It seams clear that the issue of making a switch fault-tolerant has to be reflected in the architectural design. Thinking towards the implementation therefore requires changes in both hard- and software. Since a failure of the switch hardware itself causes at least a temporary communication shutdown on the Ethernet segment, redundancy has to be introduced at his single point of failure. First, redundancy can be achieved by conventional means of device design, e. g., doubling the power supplies and therefore eliminating the most common failure. Second, any further redundancy can be obtained by a hot standby architecture in the switch hardware, as shown in Figure 2.

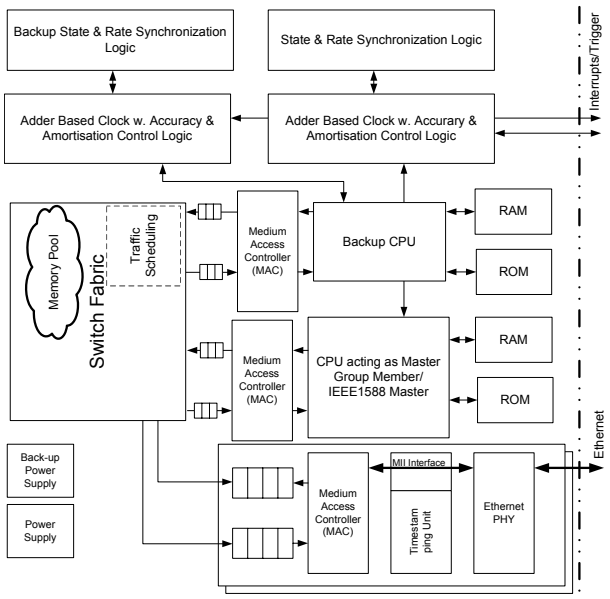

Figure 2. A fault-tolerant, IEEE 1588 and master group capable switch

Another problem is the typically single-line communication of 100 Mbit Ethernet. Since the communication cannot be reasonably enhanced by doubling the Ethernet physical layer, a redundancy introduction at this point seems to make no sense. Nevertheless, if this is a crucial issue, the transmission channel itself could be doubled by the introduction of a second communication link. In fact, backbone networks for high reliability applications depend on these principles [13]. In these architectures switches are networked with two rings, each connected to all switches. In case of failure in one ring, traffic can be routed by using the second one. Nevertheless, since the switches are allowed to decide on their own in which direction packets may be transmitted, the delay (which is in this case proportional to the hop-count) can differ from packet to packet. Thus, if these approaches are used for real-time applications, mechanisms are needed to ensure a deterministic transmission delay (at least a notification about the transmission direction in the rings). Figure 3 shows the proposed switch architecture which has a modified topology for the backbone network. Since this backbone network in state-of-the-art installations

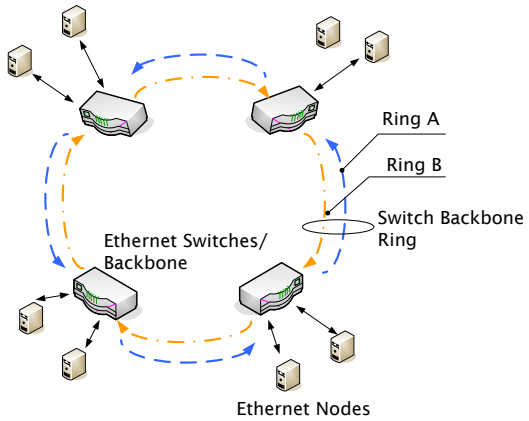

Figure 3. Inter-switch ring network [13]

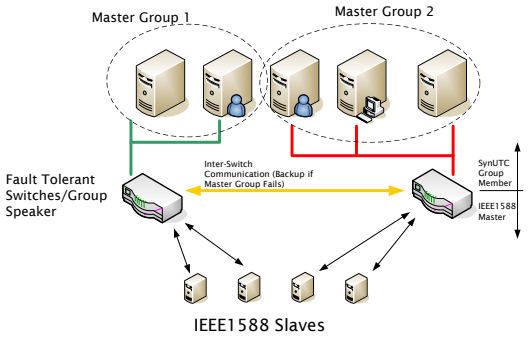

Figure 4. Connection setup to provide backup if one master group fails

is usually a fibre network anyway, the ring implementation needs no additional jacks for the two backbone connections of each node.

Another functionality of the switch is the group speaker and the democratic master group member behaviour, which is needed for the clock synchronization itself. In order to allow this, the group speaker has to act as member of the master domain and as a standard IEEE 1588 master synchronizing all slaves below. Of course the implementation of this group-speaker functionality needs a modified switch hardware. This drawback is put into perspective due to the fact that a high accuracy switch has to fulfil a second function in order to provide a high accurate clock synchronization of a Ethernet network. This second function is the so-called delta-time-stamping, where every incoming and outgoing packet is time-stamped in order to eliminate all non-deterministic delays within the switch [14].

4.3 Scalability

Another often stated request for high reliable and fault-tolerant real-time Ethernet systems is scalability. The proposed solution allows to join different master groups by just connecting their switches and therefore to implement a scalable clock-synchronized (real-time) network. In case of failure of a whole group the corresponding switch is allowed to use the other, not directly connected master group to be set as its own master group domain and again synchronize the client applications. Figure 4 shows this case including the communication between the two switches as a backup if one of the two groups fails.

5 Conclusion and Further Work

This article pointed out a new approach to enhance master-slave IEEE 1588 networks with democratic reference clocks. The goal is to increase the fault tolerance, which is typically better with fully democratic clock synchronization strategies. Since democratic approaches require a link between each network node, a tradeoff between communication overhead and robustness of the network must be made. This paper suggests a new, hybrid architecture where an ensemble of reference clocks is synchronized democratically in a so-called master group. The master group acts transparently to all IEEE 1588 nodes as a single master. Consequently the new benefit is that the failure of F out of $2F + 1$ nodes can be detected.

The approach presented in this paper is not limited to IEEE 1588 networks, and in oder to gain more generic results research regarding hybrid clock synchronization in non-IEEE 1588 networks has to be done as well. For the special case of Ethernet also the network infrastructure needs consideration because commonly available switches are a single points of failure. One way to cope with this problem is a modified switch which implements fault tolerance in addition to conventional device design techniques like doubling the power supply and providing backup processors on a hot-standby basis. Also a modified two-ring architecture of Ethernet can be provided to assure a backup if the inter-switch communication fails. Further work will deal with improved algorithms to include the master group transparently into the IEEE 1588 standard as well as a proof of concept in both simulation and hardware.

References

[1] Paolo Verissimo and Luis Rodrigues. *Distributed Systems for System Architects*. Kluwer Academic Publishers, 2001.

[2] J. Jasperneite, K. Shehab, and K. Weber. Enhancements to the time synchronization standard ieee-1588 for a system of cascaded bridges. In *Proceedings of the 2004 IEEE International Workshop on Factory Communication Systems*, pages 239–244, 2004.

[3] Martin Horauer, Klaus Schossmaier, Ullrich Schmid, Roland Höller, and Nikolaus Kerö. PSynUTC - Evaluation of a High Precision Time Synchronization Prototype System for Ethernet LANs. In *Proceedings of the Conference on Precise Time and Time Interval (PTTI)*, page 77, 2002.

[4] Hermann Kopetz. *Design principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997.

[5] Ullrich Schmid and Klaus Schossmaier. Interval-based clock synchronization. In *Journal of Real-Time Systems*, volume 2, pages 173–228, March 1997.

[6] Christof Fetzer and Flaviu Cristian. An optimal internal clock synchronization algorithm. In *Proceedings 10th Annual IEEE Conference on Computer Assurance*, Gaithersburg, MD, June 1995.

[7] Christof Fetzer and Flaviu Cristian. Integrating external and internal clock synchronization. *J. Real-Time Systems*, 12(2):123–172, March 1997.

[8] Ulrich Schmid, Martin Horauer, and Nikolaus Kerö. How to distribute GPS-time over COTS-based LANs. In *Proceedings of the 31th IEEE Precise Time and Time Interval Systems and Application Meeting (PTTI'99)*, Dana Point, California, December 1999.

[9] David L. Mills. Internet time synchronization: the network time protocol. In *IEEE Transactions on Commmunications*, volume COM-39, pages 1482–1493, October 1991.

[10] Ulrich Schmid. Synchronized UTC for distributed real-time systems. In *Proceedings 19th IFAC/IFIP Workshop on Real-Time Programming (WRTP'94)*, pages 101–107, Lake Reichenau, Germany, 1994.

[11] Paulo Verissimo, Luis Rodrigues, and Antonio Casimiro. Cesiumspray: a precise and accurate global time service for large-scale systems. *Real-Time Syst.*, 12(3):243–294, 1997.

[12] F. Cristian. Probabilistic clock synchronization. *Distributed Computing*, 3(3):146–158, 1989.

[13] Ziwen Lian, Wen-De Zheng, Kumar Bose, and Yixin Wang. Resilient ethernet ring for metropolitan area networks. In *Proceedings of the Ninth International Conference on Communications Systems*, pages 316–320, 2004.

[14] Georg Gaderer, Roland Höller, Thilo Sauter, and Hannes Muhr. Extending IEEE 1588 to fault tolerant clock synchronization 2004 IEEE International Workshop on Factory Communication Systems. In *Proceedings of the 2004 IEEE International Workshop on Factory Communication Systems*, pages 353–359, 2004.