TTEthernet Dataflow Concept

Wilfried Steiner Günther Bauer
Chip IP Design
TTTech Computertechnik AG
forename.surname@tttech.com

Brendan Hall Michael Paulitsch Srivatsan Varadarajan
Advanced Technology, Aerospace
Honeywell
forename.surname@honeywell.com

***Abstract*—*TTEthernet* is a novel communication infrastructures that allows using a single physical communication infrastructure for distributed applications with mixed-criticality requirements, e.g. the command and control systems and audio/video systems. This is achieved via a fault-tolerant self-stabilizing synchronization strategy, which establishes temporal partitioning and, hence, ensures isolation of the critical dataflows from the non-critical dataflows.**

The focus in this paper is on the dataflow in *TTEthernet*. For this we take the synchronization as a given and discuss from a *TTEthernet* user perspective which communication options *TTEthernet* provides and how they are aligned and realized. While this paper uses *TTEthernet* as reference communication infrastructure, the methods and strategies presented are valid for any message-based network.

Keywords: mixed-criticality systems, cross-industry application, Ethernet, Avionics Full-Duplex Switched Ethernet (AFDX), Time-Triggered Protocol (TTP), TTEthernet

I. Introduction

Our modern life relies on the proper function of electronic devices that provide convenience and safety. To further advance our life's quality, technology fosters the interaction of electronic devices and, hence, the communication infrastructure becomes ever more important and has to fulfill demanding requirements.

The difficulty of the interconnection is a result of requirements that not only grow with respect to the available bandwidth, but, even more complex, that are driven by real-time communication performance and fault tolerance. This mix of manifold requirements demands a new integrative communication infrastructure to keep the number of physical networks reasonable and bandwidth efficiency sustainable. In this paper we propose *TTEthernet* as such an integrative communication infrastructure that supports communication among applications with different real-time and safety requirements over a single physical network. *TTEthernet* is an industrial development which builds on the fundamental academic TT-Ethernet technology [2].

TTEthernet only uses standard Ethernet compliant frames and, hence, remains fully backward compatible to the Ethernet standard. While, in principle, any message-based communication protocol would be suited for an integrated communication infrastructure, Ethernet has significant benefits: Ethernet is the dominating standard in office communication, and, furthermore, we observe a market trend to apply Ethernet for distributed control systems. New application areas of Ethernet are, for example, industrial control, automotive, or avionics. Coming from the avionics area, *TTEthernet* was also designed to be compliant to the Avionics Full-Duplex Switched Ethernet (AFDX) [1] standard. AFDX itself is a successful attempt to increase the temporal quality and reliability of standard Ethernet. With *TTEthernet*, this technology is improved to be even usable for tight control loops.

TTEthernet provides three traffic classes: time-triggered Traffic (**TT**), rate-constrained (**RC**) Traffic, and best-effort Traffic (**BE**). Rate-constrained (**RC**) Traffic is commonly known as the traffic provided by AFDX.

This paper is structured as follows: in the next section we revise general dataflow characteristics and *TTEthernet* concepts. In Section III we elaborate qualitatively, how the *TTEthernet* traffic classes meet the general characteristics of dataflow. Section IV discusses the possible integration options, pros and cons, and the particular integration strategy chosen for *TTEthernet*. This paper concludes in Section V.

II. Dataflow Characteristics

The network discussed below is based on a switched Ethernet model. Hence, the network topology comprises a star structure consisting of end systems that are connected via a single centralized switch device or several devices that are directly connected (i.e. multi-hop communication). The channels can be implemented redundantly with respect to the degree of fault tolerance that has to be provided. The communication channel is assumed to be full-duplex, meaning that an end system is able to transmit and receive concurrently.

To maintain compatibility with existing standards, the message format utilized for *TTEthernet* is made compliant with the legacy network standard e.g. ARINC664. This standard uses Ethernet multicast MAC destination addresses as unique identifiers for all messages within an application dataflow, and this same message identifier is used in *TTEthernet* to identify a message's traffic type. In this document we use the terms message and frame interchangeably.

As depicted in Figure 1, the time-triggered services (TT Services) can be viewed in parallel to the common OSI layers: a communication controller that implements the TT Services is able to synchronize its local clock with the local clocks of other communication controllers and switches in the system. The communication controller can then send messages at off-line planned points in this synchronized global time. These messages are said to be time-triggered and it is the task of

the off-line planning tool to guarantee that the time-triggered message schedule is free of conflict. By conflict-free we mean that it will never be the case that two time-triggered messages compete for transmission and, hence, no dynamic arbitration actions for the communication medium (for time-triggered messages) are required.

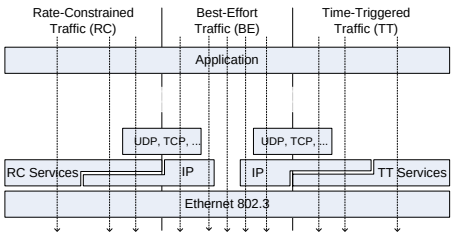

Fig. 1. Interaction of standards

As depicted in Figure 1, messages from higher layer protocols, like IP or UDP, can easily be "made" time-triggered without modifications of the messages' contents itself. This is because the *TTEthernet* protocol overhead is transmitted in dedicated messages, called *Protocol Control Frame*s, which are used to establish system-wide synchronization upon those components that need to be synchronized. In short, *TTEthernet* is only concerned with "when" a data message is sent, rather than with specific contents within a data message.

A *dataflow* defines a, possibly infinite, sequence of messages between end systems. Each dataflow has exactly one source, which we call the sender and may have an arbitrary number of end systems that consume the respective messages; we call these end systems the receivers.

A *traffic class* is the sum of all dataflows that adhere to either one communication paradigm: **TT**, **RC**, or **BE**.

III. Discussion of *TTEthernet* Traffic Classes

A. Time-Triggered Communication

TT denotes the traffic that is transmitted according to a time-triggered communication paradigm. Typically, **TT** communication requires synchrony between the time-triggered senders. In order to establish and maintain synchrony *Protocol Control Frame*s are exchanged between the end systems. *Protocol Control Frame*s are also standard Ethernet compliant frames.

Each end system stores the message dispatch points in time for **TT** messages in a local table. The reception of **TT** messages is based on the message identifier. The schedule tables are generated off-line and need to be guaranteed to be collision-free.

The problem of generating the scheduling is generally known as the scheduling problem and is aggravated when latency requirements are tight and system utilization is high. Efficient utilization requirements of hardware often requires combined solving of processing dispatch schedules and bus schedules.

1) Temporal Quality: The pre-planned message exchange is one of the most important benefits of time-triggered communication: timeliness is guaranteed and latency as well as jitter can be kept small as contentions for the shared medium are avoided.

2) Protocol Overhead: Time-triggered communication relies on the implementation of synchronization services. Such services include a method to establish an initial agreement on the values of the local clocks of the components in the system (i.e. a startup algorithm [6]) and a clock synchronization service [3] that maintains the synchronization of the local clocks during normal operation mode. In addition, a time-triggered communication protocol supports services for restart in case the synchronization is lost. Synchronization services are distributed algorithms that involve the cooperation of a set of end systems and switch(es) in the network. These services need to be fault-tolerant. The *TTEthernet* synchronization services are defined in [5].

3) Composability/Scalability: A time-triggered communication paradigm requires locally stored schedule tables. If new messages have to be scheduled, e.g. when adding a new end system, these schedule tables have to be modified. To reduce the impact of such changes the tables are organized to support an "incremental scheduling philosophy", which is a method that allows building a new schedule from an existing one without changing the required latency constraints.

B. Rate-Constrained Communication

RC denotes the traffic that is transmitted according to a rate-constrained communication paradigm. Unlike **TT**, **RC** traffic does not rely on *Protocol Control Frame*s. **RC** is the communication paradigm realized by ARINC 664 part 7 (Avionics Full-Duplex Switched Ethernet). Each sender realizes a traffic shaping function for each **RC** dataflow. This traffic shaping function ensures that there will be a minimum inter-frame gap between any two following messages of a dataflow. Each switch in the network realizes a traffic checking function that checks whether, indeed, the end systems produce a well-shaped sequence of messages. A switch drops messages that are sent too early, hence, violating the minimal inter-frame gap.

1) Temporal Quality: The rate of message transmission of **RC** traffic is bounded. As the end systems are not synchronized, peak-load scenarios have to be considered to calculate the required buffer depth, such that no message is lost. As the rates and the buffer sizes are a priori given, the latency and jitter of a message can be calculated at design time and message transmission can be guaranteed. However, this static analysis may disclose that latency and jitter are unacceptable. In this case, a probability analysis can be performed that results in a stochastic distribution of a message's temporal quality.

2) Protocol Overhead: For **RC** traffic the end system has to realize a traffic shaping function and the switches have to realize a traffic checking function. If an end system is the source of multiple dataflows the traffic shaping function has to

be done individually. Furthermore, services as the redundancy service and the integrity service have to compensate for the low temporal and deterministic qualities.

3) Composability/Scalability: Adding a new component to the system requires a recalculation of the performance parameters involving the overall system, such that message transmission is still guaranteed. This includes a recalculation of the buffer sizes in the switch and an evaluation of whether the latency guarantees are still met. Of course, similar techniques as incremental scheduling for **TT** can be applied, by reserving bandwidth for future extensions when the initial performance parameters are calculated.

C. Best-Effort Communication

BE denotes the traffic that is transmitted according to a best-effort communication paradigm.

1) Temporal Quality: No temporal guarantees can be given, as message drops in the switch are possible, if the network buffers are overflown or messages will never be transmitted as there is not enough bandwidth available.

2) Protocol Overhead: While the mandatory services for **BE** traffic are basically a transmission and a reception service, there are several optional services that can be implemented for **BE**: address learning in the switch, back-pressure algorithm (e.g. PAUSE command in Ethernet), message filtering based on message identifier, retransmission service in case a message is lost, etc.

3) Composability/Scalability: AS **BE** gives neither temporal nor deterministic qualities, the impact of adding a new component cannot be estimated. As all end systems compete for the available bandwidth, the dataflow of already integrated end systems can be affected.

IV. Dataflow Integration

TTEthernet supports dataflows of all three traffic classes (**TT**, **RC**, and **BE**). This section discusses the mechanisms and consequences when integrating these dataflows onto a single physical network.

A. Integration Methods

When unsynchronized dataflows are integrated, typically, contentions occur on the outgoing ports of a switch and/or end system. If the contention occurs to messages of dataflows with equal priority, these messages will be served in FIFO. Also, when a message of a high-priority dataflow (**H**) is served and a low-priority message (**L**) becomes ready, **L** will be queued. The third case, though, is of particular interest: we consider three different integration methods to resolve contentions when a low-priority (**L**) messages is already in transmission and a high-priority (**H**) messages gets ready. The three methods are: preemption, timely block, and shuffling and are depicted in Figure 2.

1) Preemption [2]:

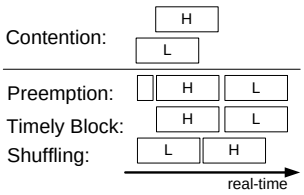

Fig. 2. Integration methods for high-priority (H) and low-priority (L) traffic

a) Description: If an **L** message is relayed by a switch when a **H** message arrives, the relay process of the **L** is stopped. The switch establishes the minimum time of silence on the channel and relays the **H** message an a priori specified duration later.

b) Pros: **High Real-Time Quality:** Preemption guarantees that the switch introduces an almost constant and a priori known latency for a **H** message.

c) Cons: **Generation of False Messages:** As truncated messages are now systematically generated by this mechanism, it has to be guaranteed that a truncated message does not appear to a receiver as a correct message. This can be ensured by the preemption mechanism to generate a signal pattern that violates the line encoding rules when a message is truncated or to include the original message length within the message. Hence, all receivers of this truncated message will identify a syntactically faulty message.

Resource Inefficiency: If standard Ethernet components are to be supported, each truncation action results in a loss of bandwidth, as the already transmitted fraction of a message is lost in a receiver and the whole message has to be retransmitted. Add-on functionality to standard Ethernet that allow reconstruction of fragmented Ethernet messages at a receiver [4] allows allows re-claiming, at least parts of, the lost bandwidth.

2) Timely Block:

a) Description: If the **H** message is a **TT** message, the switch in the network knows a priori when this **H** message will arrive on which port, and to which ports (or an internal buffer) this message has to be forwarded. Timely Block means that the switch will not forward messages at those times when a **TT** message is expected.

b) Pros: **High Real-Time Quality:** As they outgoing ports for an **H** message are scheduled to be free when the **H** message arrives, the integration-imposed delay is almost constant.

c) Cons: **Resource Inefficiency:** When the message lengths of the **L** messages are not known, the timely block has to be at least of a duration of the maximum possible **L** message. In case of Ethernet this means either $123.040\mu s$ for 100 Mbit/sec or $12.304\mu s$ for 1 Gbit/sec. As the minimum sized Ethernet frames are $6.72\mu s$ for 100 Mbit/sec and $0.67\mu s$ for 1 Gbit/sec, the timely block may cause up to 19 Ethernet frames that cannot be delivered. To overcome this resource inefficiency the switch/end system can act more intelligently, when the message lengths of the **L** messages are known or

transported in the messages themselves: at any point in time when an **L** message is ready to be relayed, the switch will only relay this **L** message if it is guaranteed that the **L** message is completely relayed before the **H** message has to be relayed.

3) Shuffling:

a) Description: If an **L** message is relayed by a switch when an **H** message arrives, the **H** message is delayed until the relay process of the **L** message is finished. Hence, in the worst case the **H** message is delayed for a maximum sized **L** message. This delay will also impact following **H** messages until the bandwidth required for the **L** message is compensated by the sum of the inter-frame gaps between two succeeding **H** messages.

b) Pros: **Resource Efficiency:** In contrast to preemption and timely block, shuffling will not truncate a message, nor block the outgoing ports for **L** messages. Hence, from a utilization point of view, shuffling is an optimal solution.

c) Cons: **Low Real-Time Quality:** If the **H** message is a **TT** message, the good real-time quality of time-triggered traffic is degraded. However, as time-triggered traffic is dispatched according to a synchronized timbase, this dispatch point in time is *a priori* known to the receivers. Messages of other traffic classes can include a global timestamp within the message to inform the receivers of their dispatch point in time. Hence, the synchronized timebase mitigates the increased transmission jitter. Network latency, on the other hand, cannot be mitigated: an **H** message will potentially be delayed by at most one **L** message at each communication step. However, in a 100 Mbit/sec network, or 1 Gbit/sec network, shuffling still gives sufficient real-time quality for a broad range of in-vehicle applications, such as avionics or automotive applications.

Interference with BE Traffic: As an **H** message may be delayed by an **L** message in general, this is also true in particular for those cases where **L** is a **BE** message. Though the influence of **BE** messages is bounded and guaranteed by *TTEthernet* shuffling may not be acceptable in applications where **BE** messages are sent from uncontrolled sources, e.g. a flight passenger connecting their laptop to a cabin network, where also alarm data is communicated.

Increased Complexity in Composability/Scalablity for TT: Shuffling requires constraints on the scheduler of **TT** messages: The increased latency and jitter affects the communication schedule. Incremental scheduling is only successful if changes of the **TT** timing (due to **RC** and **BE** messages) is taken into account when creating the initial communication schedule. However, there will be no issue for the application and task-level scheduler if these constraints are taken into account, right from the beginning.

B. Integration Strategy

The first *TTEthernet* products realize the timely block and shuffling integration method. The preemption integration method is available in a prototype state. For the integration strategy we define a priority scheme in decreasing order: *Protocol Control Frame*s, **TT** dataflows, k priorities for **RC** dataflows, and **BE** dataflows. Figure 3 presents an example TTEthernet network consisting of three end systems and a single switch.

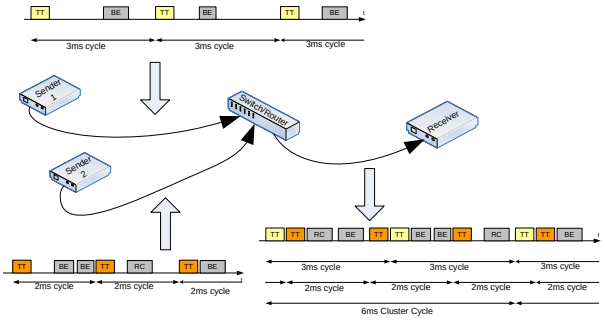

Fig. 3. TTEthernet – Example of dataflow integration

The example consists of two end systems that send frames, a switch that integrates the frames from the two senders, and a receiver that receives the integrated dataflow from the switch. As depicted, sender 1 sends a time-triggered frame (TT) with a period of 2 milliseconds and best-effort frames (BE). Sender 2 sends a time-triggered frame (TT) with a period of 3 milliseconds, best-effort frames (BE), and rate-constrained frames (RC). The resulting integrated dataflow is depicted on the right.

V. Conclusion

In this paper we presented the concepts and rationale behind the dataflow options in a *TTEthernet* network. These concepts have been realized and validated in numerous prototype generations and are available as products, which have been selected for industrial and avionics applications.

Disclaimer

This paper solely represents the personal opinion of the listed authors. Under no circumstances the material presented in this paper can be credited as an official statement of one or many affiliations of the authors. Features and functionality of *TTEthernet* may vary depending on particular products.

The research leading to these results has received funding from the [European Community's] Seventh Framework Programme ([FP7/2007-2013]) under grant agreement n° [236701] (*CoMMiCS*).

References

[1] AEEC, *ARINC PROJECT PAPER 664, AIRCRAFT DATA NETWORKS, PART7, AFDX NETWORK (DRAFT)*, AERONAUTIC RADIO, INC., 2551 Riva Road, Annapolis, Maryland 21401-7465, Nov. 2003.

[2] H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer, "The time-triggered ethernet (tte) design," *8th IEEE International Symposium on Object-oriented Real-time distributed Computing (ISORC), Seattle, Washington*, May. 2005.

[3] L. Lamport and P. M. Melliar-Smith, "Synchronizing clocks in the presence of faults," vol. 32, no. 1, pp. 52–78, Jan. 1985.

[4] V. Mikolasek, A. Ademaj, and S. Racek, "Segmentation of standard ethernet messages in the time-triggered ethernet," Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-1, 1040 Vienna, Austria, Tech. Rep. 22/2008, 2008.

[5] W. Steiner, *TTEthernet Specification*, TTA Group, 2008.

[6] W. Steiner and H. Kopetz, "The startup problem in fault-tolerant time-triggered communication," *International Conference on Dependable Systems and Networks (DSN 2006)*, Jun. 2006.