Presented at the IEEE World Forum on Internet of Things (WF-IoT), IEEE, 2014, pp. 433-436

Towards Synchronous Deterministic Channels for the Internet of Things

Wilfried Steiner
TTTech Computertechnik AG
wilfried.steiner@tttech.com

Flavio Bonomi
IoXWorks, Inc.
fgbonomi@gmail.com

Hermann Kopetz
Vienna University of Technology
hermann.kopetz@tuwien.ac.at

***Abstract*—Embedded systems are omnipresent in our daily lives and rapidly gaining more and more importance, with performance improvements and technological evolutions in several directions. One of the most promising directions of evolution is in their interconnection. Already today there is a broad variety of use cases for interconnected embedded systems (automobiles, airplanes, automated production halls, etc.) and the most cutting edge examples demand the distributed embedded systems to tightly couple their operation. As a direct result of this tight coupling, stringent requirements on the underlying communication network immediately arise.**

The Internet-of-Things (IoT) as a discipline and movement to systematically connect distributed embedded systems on large or even Internet-like scales, promises a new generation of systems and system-of-systems. In order to also enable tight coupling of embedded systems in the IoT, there is a need for deterministic communication channels with predictable and low communication latency and jitter. In this paper we discuss how synchronous deterministic channels can be developed for the IoT by application of the time-triggered communication paradigm and other technology standards currently under development.

I. Introduction

Today, embedded systems are a corner stone of society - their ubiquitous presence and increasing relevance in daily-life products, vehicles, infrastructures, or factories, just to name a few, defines to a large degree how we perceive technology as a whole and how technology improves the quality of our lives. Furthermore, in our high-technology affected world, innovators and engineers quickly and steadily develop new and novel use cases for embedded systems, for example novel mobility concepts, smart grid technologies, or assistance systems for elderly care. New means for the inter-connectivity of embedded systems such as the Internet of Things (IoT) promise a significant leap forward for the embedded systems industry and the services that embedded systems will provide to society.

Embedded systems are omnipresent in our daily lives, and are becoming increasingly large and complex. As a consequence of this trend it is apparent that the correct development of such complex systems requires a sound architectural basis. In the absence of architectures we will either build systems of insufficient quality or will simply not be able to build systems beyond a certain level of complexity at all. The time-triggered architecture (TTA) [1] as developed at the Institut für Technische Informatik at the Vienna University of Technology is an extraordinary example of an architecture for distributed embedded systems. The TTA tremendously simplifies the development of distributed embedded systems such as cyber-physical systems. It has been successfully applied in industries that demand a high level of determinism such as the avionics industry, in which predictability of system operation is a key property. TTP [2] and TTEthernet [3] are implementations of the TTA. TTP is applied, for example, in the new Boeing 787 Dreamliner, whereas TTEthernet has been selected for the NASA Orion Space Program. While the aerospace and space industries (as well as automotive and similar industries) are traditional areas for dependable embedded systems, we also observe emerging areas with increasing dependability requirements. Examples include surgical robots in the medical area, datacenters in critical industries, as well as the smart grid that aims for decentralized energy production and efficient energy use. TTEthernet is currently being evaluated for several of these areas.

The transfer of the capabilities provided by a time-triggered architecture to the IoT enables a large variety of novel functionality as it allows to geographically decouple computational expensive processes from the location of action. E.g., robots can be tightly controlled from centralized stations or automotive driver-assistance systems can connect to the cloud to improve their services.

In the next section we give an overview of the TTEthernet technology as an example of how to enable synchronous deterministic channels in a network. We will focus in particular on the need for robust clock synchronization and the integration of traffic classes with different quality of service transmission guarantees. We will then discuss technological challenges and recent advancements to bring the time-triggered paradigm to large-scale systems and Internet-like systems like the IoT. We will start with clock synchronization in Section III and continue with communication-scheduling issues in Section IV. Finally, we conclude in Section V.

II. Synchronous Deterministic Channels

Two fundamental mechanisms need to be in place to realize synchronous deterministic channels: first the components that communicate using these channels need to synchronize their local clocks and, secondly, the components need to agree on an access pattern of the channel (called a communication schedule).

TTEthernet (SAE AS6802, [4]) is an example of how to establish synchronous deterministic channels in a layer 2 network. It is an extension to standard Ethernet for usage in safety-related networks and cyber-physical systems such as avionics, automotive applications, or green energy generation systems. An example TTEthernet network is depicted in Figure 1. It consists of eight end systems connected to each other via two redundant channels (a top and a bottom channel). Each of the channels is composed of three TTEthernet switches which are connected to each other with multi-hop links.

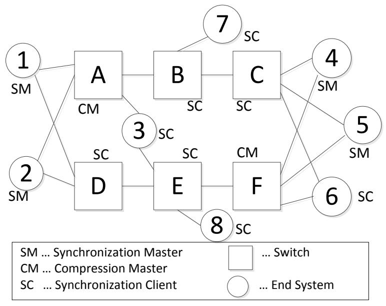
1
SM
2
SM
A
CM
B
SC
C
SC
D
SC
E
SC
F
CM
3
SC
4
SM
5
SM
6
SC
7
SC
8
SC
SM ... Synchronization Master
CM ... Compression Master
SC ... Synchronization Client
... Switch
... End System

Fig. 1. Example TTEthernet network

In standard Ethernet, messages are communicated according to a best-effort paradigm, which means that no bounds on a message's transmission latency can be given. Under high load buffers in network switches can overflow and messages are lost entirely or delayed for an unacceptable time when using back-pressure flow control. In noncritical systems, higher-layer protocols such as TCP/IP often compensate for message loss using re-transmission strategies. However, networked cyber-physical systems typically have stringent end-to-end timing requirements that do not allow the temporal penalty of repeated transmission attempts. Furthermore, standard networks cannot guarantee that any successive transmissions will actually be successful.

TTEthernet achieves timely transmissions with known upper bounds on latency and jitter by providing rate-constrained (RC) and time-triggered (TT) communication paradigms in addition to standard best-effort (BE) traffic. An example communication scenario showing TT, RC, and BE traffic is depicted in Figure 2. It describes end system 1 transmitting a TT message (TT1), an RC message (RC1), and a BE message (BE1) through switches A and B to end system 7. Furthermore, end system 2 also transmits a TT message (TT2) through switches A and B to end system 7.

Rate-constrained traffic is well-known from an avionics Ethernet variant (ARINC 664P7-1 [5]). It is based on an *a priori* agreement of the network components on the number, size, and maximum frequency of messages. This knowledge is sufficient to calculate the required network resources, i.e., the switch buffers, and it can be guaranteed that no message will be lost due to buffer overflows even in absence of explicit flow-control protocols. However, different network components may source their messages at about the same point in time or may even source several messages back to back. This uncoordinated transmission pattern quickly leads to

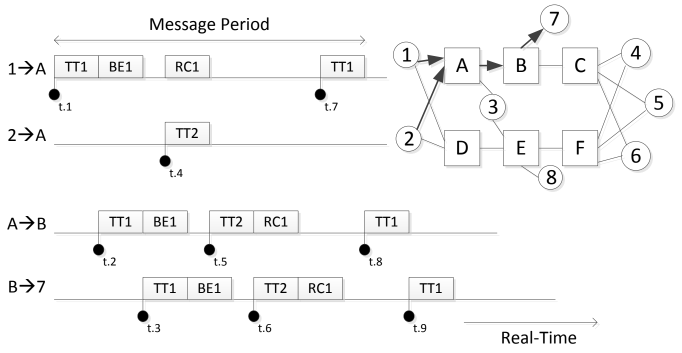

Fig. 2. Example TTEthernet communication scenario

high transmission latencies. TTEthernet also support the IEEE 802.1 AVB (audio/video bridging) standard which specifies a similar rate-constrained communication paradigm.

The time-triggered communication paradigm takes the level of determinism to the extreme. Here, all communication participants are equipped with local clocks that are brought in close agreement to establish a synchronized global time. An Ethernet frame may now be dispatched at an *a priori* specified point in the global time, which makes the frame transfer time-triggered (in Figure 2 these points in time are depicted by the black dots labeled t.1 to t.9). The sum of all those specified points in time for dispatch, relay, and potentially also reception is collectively referred to as the "communication schedule" for time-triggered communication. When correctly aligned, the communication schedule ensures that any two time-triggered Ethernet frames will never compete for transmission resources (e.g., communication links, switch buffers) and their transmission latencies can be kept minimal and almost constant.

For time-triggered communication, the local clocks of the end systems and switches need to be synchronized to each other. TTEthernet standardizes a fault-tolerant clock synchronization protocol in SAE AS6802 that guarantees this synchronization even in the event of faulty components. The clock synchronization protocol specifies three different synchronization roles: synchronization master, synchronization client, and compression master. Typically some of the end systems are configured as synchronization masters and one switch per channel in the network is configured as compression master. Components that are neither synchronization master nor compression master act as synchronization clients and only passively synchronize to the synchronized timebase as established and maintained by the synchronization and compression masters. In the example TTEthernet network in Figure 1 end systems 1, 2, 4, and 5 are configured as synchronization masters (SM), and switch A and F act as compression master (CM). All other components are synchronization clients (SC). During operation the synchronization masters periodically send synchronization messages to the compression masters which will perform a specialized form of fault-tolerant average on the time values that the synchronization messages from the synchronization masters represent. The compression masters will then send the result of the fault-tolerant average back to the synchronization masters and synchronization clients which will then, again, execute an averaging function on the time values associated with the synchronization messages sent by the compression masters. Hence, the clock synchronization protocol in TTEthernet consists of two convergence steps, a first convergence is executed in the compression masters and a second convergence is executed in the synchronization masters and clients. More details on the operation and the formal verification of the TTEthernet synchronization protocols can be found in [6].

To achieve highly available message delivery, TTEthernet supports the parallel communication of messages via all redundant channels. For example, the top left end system in Figure 1 may need to communicate with the end systems on the right with guaranteed message delivery. In this case the end system will send its messages via both channels (top and bottom) in parallel. The end stations on the right execute a redundancy management service and deliver only one of the redundantly communicated messages to the applications that they host (or both copies if the redundancy management is configured to do so). Hence, even in case of the failure of any one switch or communication link in the system it is guaranteed that the communication is successful and at least one copy of each message sent will also be received by the end stations.

There are several technical challenges to be addressed in order to scale deterministic channels, and synchronous ones in particular, to an Internet-like dimension. An example of such a large-scale network is depicted in Figure 3. As shown in this scenario, the IoT connects several layer 2 networks via a layer 3 network. For fault-tolerance and robustness reasons, the layer 2 networks are connected with redundant communication links into the layer 3 cloud. In such a setting, synchronous deterministic channels for the IoT not only need to be established between end systems within the same layer 2 network, but also connecting devices in different layer 2 networks, connected to each other via a layer 3 network.

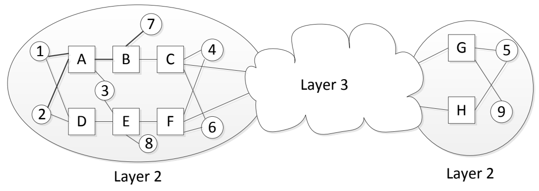

Fig. 3. Large-scale deterministic network using layer 2 and layer 3

Following the two fundamental mechanisms necessary for synchronous deterministic channels, we first discuss these hurdles and challenges for clock synchronization followed by the communication scheduling mechanism.

III. Clock Synchronization Challenges and Recent Developments

The synchronization of local clocks even in large-scale networks is not a technical problem *per se*. There are several clock synchronization protocols already widely used today, e.g., the network time protocol (NTP) [7] or IEEE 1588/802.1AS [8], [9]. However, the technical challenge of clock synchronization comes with the quality of synchronization (QoSync) that needs to be achieved. Some quality aspects of synchronization are as follows:

- precision: worst-case difference of any two non-faulty clocks in the system
- accuracy: worst-case difference of the clocks in the system to an external time reference
- startup time: worst-case time after startup of the time sources until the system is synchronized (with given precision and/or accuracy)
- integration time: worst-case time for a non-synchronized component in the system to become synchronized
- changeover time: worst-case time for the components in the system to change from one time source to another one (e.g., in the case that the original time source fails)
- recovery time: worst-case time for the synchronized time-base to recover after global synchronization loss

In large-scale networks, protocols like NTP and IEEE 1588/802.1AS may achieve a precision in the range of several milliseconds down to some microseconds (and sometimes below). However, to the best knowledge of the authors these numbers only cover the average case and not the worst case of operation. For example, the performance of IEEE 1588/802.1AS is studied by means of simulation [10] rather than by means of formal methods or deductive proof. TTEthernet on the other hand has been studied by formal methods in order to get performance characteristics of the clock synchronization protocol under worst-case system behavior including faulty components. Ideally, we could combine the different protocols NTP, IEEE 1588/802.1AS and TTEthernet (SAE AS6802) to achieve a robust large-scale synchronization protocol with known and acceptable QoSync.

Indeed there is an ongoing standardization effort in the IEEE 802.1 group to enhance IEEE 802.1AS towards more robustness. One way that is currently proposed and discussed is to install a TTEthernet (SAE AS6802) system as a high-availability time source for IEEE 802.1AS [11]. The same concept will be presented also the the IEEE 1588 community. Furthermore, we are currently also looking into how to couple Global Positioning System (GPS)-based clocks to a TTEthernet fault-tolerant time-base. In general the growing importance and availability of a global time-base is acknowledged by organizations like the consortium around "The Network and Information Technology Research and Development (NITRD) Program", which recently held a workshop on "The New Clockwork for Time-Critical Cyber-Physical Systems" [12]. Hence, despite the technological as well as societal hurdles, we observe international research trends indicating that it is realistic to assume that a high-quality global time-base will become more and more accessible to the world. The IoT should take advantage of this global time-base to realize synchronous deterministic channels.

IV. Challenges and Recent Developments in Scheduling

The second challenge in developing synchronous deterministic channels for the IoT are new methods and means to establish communication schedules. Yet, again, scheduling *per se* is not a challenge, but rather establishing a certain quality of a schedule is (e.g., configuring a deterministic channel for a single message will likely not be a problem at all, but configuring the same channel for thousands of message will likely be). Some of these quality aspects are as follows:

- schedule size: this includes the number of messages to be scheduled, the number of hops in between senders and receivers, the number of receivers, etc.
- schedule tightness: this quality parameter targets at how many messages need to be scheduled per given interval, i.e., how tightly need messages to be scheduled back to back.
- schedule structure: a communication schedule may require fixed time slots in which messages are communicated, or time slots of variable size.
- schedule porosity: in order to integrate non-scheduled traffic, the schedule can foresee "gaps" to be used for non-scheduled traffic. The schedule quality parameter refers to the number, size, and positions of these gaps.
- schedule heterogeneity: the schedule may need to take different network technologies, such as differing communication speeds, mixed wired/wireless communication links, or differing communication delays through switches and routers into account.
- schedule dynamics: the schedule may change during the operation time of the system, i.e., new messages may need to be scheduled for the synchronous deterministic channel, existing ones may be deleted or modified.
- schedule performance: this quality aspect targets at how fast a schedule needs to be established or updated.

From this variety of schedule quality aspects we can easily deduce that the scheduling problem can become of almost arbitrary complexity when all quality aspects need to be satisfied to highest degrees. Hence, in traditional time-triggered communication systems like TTEthernet, the required quality aspects are significantly restricted to configuration patterns, e.g., non-dynamic scheduling, same slot size for all messages, only a few regular gaps for scheduling porosity, etc. However, the traditional configuration patterns chosen, follow the requirements from traditional application areas of time-triggered communication like avionics or automotive applications.

For the IoT new configuration patterns for the synchronous deterministic communication channels need to be found and agreed between all stakeholders (e.g., within standardization consortia like IEEE and IETF). In particular for IoT it is likely that the schedule dynamics, schedule heterogeneity, and schedule performance become significantly more important than they used to be.

Indeed, the IEEE 802.1 time-sensitive networking (TSN) task group recently started to standardize on some basic time-triggered capabilities in the IEEE 802.1Qbv project. Once the project is concluded, standard Ethernet bridges (aka switches) will be able to enable and disable transmission queues based on a schedule. Furthermore, the ongoing IETF Path Computational Element (PCE) working group [13] could be a good forum to further develop configuration patterns as well as protocols and algorithms to distribute the configuration in large scale systems.

V. Conclusion

The Internet of Things is upon us and it should serve as an interconnect to tightly couple distributed embedded systems. Therefore, we have discussed how the time-triggered communication principle can be used to establish synchronous deterministic communication channels that enable such a tight coupling. We discussed clock synchronization as well as scheduling aspects and in particular issues that arise in large scale systems. Several aspects of synchronous deterministic channels have not been addressed in this paper, e.g., certification aspects or large-scale validation and testing aspects and will be addressed in future work.

Acknowledgments: The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007- 2013) under grant agreements $n°$610640 (*DREAMS*) and $n°$610535 (*AMADEOS*).

References

[1] H. Kopetz and G. Bauer, "The time-triggered architecture," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 112 – 126, Jan. 2003.

[2] H. Kopetz, *TTP/C Protocol – Version 1.0*. Vienna, Austria: TTTech Computertechnik AG, Jul. 2002, Available at http://www.ttpforum.org.

[3] W. Steiner, *TTEthernet Specification*, TTA Group, 2008, Available at http://www.ttagroup.org.

[4] W. Steiner, G. Bauer, B. Hall, and M. Paulitsch, "TTEthernet: Time-Triggered Ethernet," in *Time-Triggered Communication*, R. Obermaisser, Ed. CRC Press, Aug 2011.

[5] *Aircraft Data Network, Part 7: Avionics Full Duplex Switched Ethernet (AFDX) Network*, ARINC Report 664P7-1, Sep. 2009.

[6] W. Steiner and B. Dutertre, "Automated formal verification of the *ttethernet* synchronization quality," in *NASA Formal Methods*, 2011, pp. 375–390.

[7] D. L. Mills, "Internet time synchronization: the network time protocol," *Communications, IEEE Transactions on*, vol. 39, no. 10, pp. 1482–1493, 1991.

[8] "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pp. c1–269, 2008.

[9] J. Teener and G. M. Garner, "Overview and timing performance of ieee 802.1 as," in *Precision Clock Synchronization for Measurement, Control and Communication, 2008. ISPCS 2008. IEEE International Symposium on*. IEEE, 2008, pp. 49–53.

[10] G. M. Garner, A. Gelter, and M. J. Teener, "New simulation and test results for ieee 802.1 as timing performance," in *Precision Clock Synchronization for Measurement, Control and Communication, 2009. ISPCS 2009. International Symposium on*. IEEE, 2009, pp. 1–7.

[11] W. Steiner, "Interoperability of IEEE 802.1AS and fault-tolerant clock synchronization," http://www.ieee802.org/1/files/public/docs2013/new-avb-wsteiner-8021AS-interoperability-ft-clocksync-0913-v03.pdf, accessed: 2013-10-09.

[12] "2012 national workshop on the new clockwork for time-critical cyber-physical systems," http://cps-vo.org/group/time-criticalworkshops, accessed: 2013-10-09.

[13] "Path computation element (pce)," http://datatracker.ietf.org/wg/pce/charter/, accessed: 2013-10-09.