Optimization of TTEthernet Networks to Support Best-Effort Traffic

Tamas-Selicean, Domitian; Pop, Paul


Link to article, DOI: 10.1109/ETFA.2014.7005256

Publication date: 2014

Optimization of TTEthernet Networks to Support Best-Effort Traffic

Domitian Tămaș–Selicean and Paul Pop
DTU Technical University of Denmark
Kongens Lyngby, 2800 Denmark
Email: dota@dtu.dk, paupo@dtu.dk

Abstract—This paper focuses on the optimization of the TTEthernet communication protocol, which offers three traffic classes: time-triggered (TT), sent according to static schedules, rate-constrained (RC) that has bounded end-to-end latency, and best-effort (BE), the classic Ethernet traffic, with no timing guarantees. In our earlier work we have proposed an optimization approach named DOTTS that performs the routing, scheduling and packing / fragmenting of TT and RC messages, such that the TT and RC traffic is schedulable. Although backwards compatibility with classic Ethernet networks is one of TTEthernet's strong points, there is little research on this topic. However, in this paper, we extend our DOTTS optimization approach to optimize TTEthernet networks, such that not only the TT and RC messages are schedulable, but we also maximize the available bandwidth for BE messages. The proposed optimization has been evaluated on a space application case study.

I. INTRODUCTION

The increase in functionality that is implemented as real-time embedded applications, most often on distributed architectures, has resulted also in increased bandwidth requirements. A large number of communication protocols have been proposed for embedded systems. Although Ethernet [6] is not suitable for real-time and safety-critical applications [3], many Ethernet-based communication solutions have been proposed in recent years, due to the low costs and high speeds of Ethernet (up to 10 Gbps). Some examples are ARINC 664 Specification Part 7 (ARINC 664p7, for short) [1], TTEthernet [9], EtherCAT [4] and IEEE Audio Video Bridging (AVB). [10] compares several proposed Ethernet-based real-time communication protocols.

TTEthernet [9] is a deterministic, synchronized and congestion-free network protocol based on the IEEE 802.3 Ethernet [6] standard and compliant with the ARINC 664p7 specification [1]. ARINC 664p7 is a full-duplex Ethernet network, which emulates point-to-point connectivity over the network by defining virtual links, tree structures with one sender and one or several receivers (see Section II). TTEthernet supports applications with mixed-criticality requirements in the temporal domain, providing three types of traffic: static time-triggered (TT) traffic and dynamic traffic, which is further subdivided into Rate Constrained (RC) traffic that has bounded end-to-end latencies, and Best-Effort (BE) traffic, for which no timing guarantees are provided. TT messages are transmitted on static schedule tables and have the highest priority. RC messages are transmitted if there are no TT messages in transmission, and BE traffic has the lowest priority. TTEthernet is highly suitable for safety-critical applications of different safety criticality levels, as it offers spatial separation for mixed-criticality messages through the concept of virtual links. TTEthernet is suitable for automotive [12], avionics [15] and space [5] applications.

Several researchers have started to address the analysis and optimization of TTEthernet. For full-duplex switched Ethernet networks with priority operations, Schneider et al. [11] have proposed a compositional timing analysis based on real-time calculus [2]. For ARINC 664p7 systems, researchers [7] have proposed a new real-time switching algorithm that guarantees an upper bound on the switching period. Having such an upper bound simplifies the worst-case delay analysis. For TTEthernet, Steiner [13] proposes an approach for the synthesis of static TT schedules, where he ignored the RC traffic and used a Satisfiability Modulo Theory (SMT)-solver to find a solution which satisfies an imposed set of constraints. The same author has proposed an SMT-solver approach to introduce periodic evenly-spaced slots into the static schedules to help reduce RC delays in [14]. Suethanuwong [16] proposes a scheduling approach of the TT traffic, ignoring RC traffic, that introduces equally distributed available time slots for BE traffic.

In this paper we are interested in safety-critical real-time applications implemented on heterogeneous processing elements interconnected using TTEthernet. In [17] we have proposed an approach that focuses only on deriving the static schedules for the TT messages. In [18] we have proposed the DOTTS ("Design Optimization of TTEthernet-based Systems") strategy, a Tabu Search-based metaheuristic that optimizes the TTEthernet network implementation, such that the TT and RC messages are schedulable, and the end-to-end delay of RC messages is minimized. DOTTS optimizes: the routing of virtual links (VLs), the packing and fragmenting of messages into frames, the assignment of frames to VLs, the bandwidth of the RC VLs and the schedules for the TT messages.

DOTTS focuses only on TT and RC traffic. However, one of the strong points of TTEthernet is the backwards compatibility with Ethernet (the BE traffic). In practice, the BE traffic is used for non-critical applications, without timing or safety constraints, and for legacy components, which do not support TT or RC communications. Hence, it is imperative to accommodate the BE traffic in TTEthernet networks. There is no research, that we are aware of, that supports the integration of BE traffic together with TT and RC traffic. Hence, in this paper we propose an extension to our TT and RC optimization approach DOTTS [18] to consider also the BE traffic, such that the TT and RC messages are schedulable and the bandwidth available for BE messages is maximized.

In Section V we show how DOTTS can be modified to take into account also BE (regular Ethernet) traffic, by updating its cost function and adding new design transformations. We refer to this updated version of DOTTS as DOTTS+. We have evaluated DOTTS+ using a real-life test case.
composed of a set of dataflow paths, one such dataflow path
We assume that the designer has decided the traffic classes
message \( m \) class of each frame (TT, RC or BE). The size
A bit pattern specified in the frame header identifies the traffic
classification \([1]\). Messages are transmitted in the payload of frames.
A virtual link \( \nu \) as the root and the receivers as leafs. For example,
A virtual link
Each virtual link \( \nu \) is depicted
The space partitioning between messages of different crit-
the TT and RC messages we know their periods / rate and
deadlines, \( m_i \) period or rate, and \( m_i \) deadline, respectively.
RC messages are not necessarily periodic, but have a minimum
inter-arrival time. We define the rate of an RC message \( m_i \) as
we model BE messages as aperiodic or sporadic, with a minimum inter-arrival time (denoted by
In TT Ethernet networks, BE traffic can be either statically routed, or be routed via address
learning in the switches. In this paper, we assume that the BE
traffic is statically routed.
We compute the necessary bandwidth for a message \( m_i \) as:
In case of the aperiodic BE messages, we substitute in Eq. 1
We have shown in \([17]\) how the TT Ethernet protocol works
for TT and RC frames. Next, we will show how BE messages
are transmitted. For the sake of simplicity, we consider the
same topology as in Fig. 1. We consider that \( ES_1 \) transmits
TT message \( TT_1 \) to \( ES_1 \), and RC message \( RC_1 \) to both \( ES_1 \)
and \( ES_3 \), while \( ES_2 \) sends RC message \( RC_2 \) to \( ES_1 \), and the
BE messages \( BE_1 \) to \( BE_4 \) to \( ES_3 \). Fig. 2a depicts, using a Gantt
diagram, the transmission of frames on dataflow links \([ES_1, NS_1],[ES_2, NS_1],[NS_1, ES_3]\].
TT Ethernet has three traffic integration policies: shuffling,
preemption and timely block (described in \([17]\)). In this paper
we consider the timely block policy: a low priority frame
cannot be transmitted if it interferes with the transmission of
a scheduled TT frame. Such a timely block interval is shown
in Fig. 2a with a hatching pattern on \([NS_1, NS_2]\) between \( BE_1 \)
and \( TT_1 \). We denote with \( BW_{\text{blocked}}(l_i) \) the bandwidth taken
by the timely blocked interval around time unit 11. On the
other hand, the shuffling traffic integration policy allows lower
criticality messages to be sent even if this will result in
delaying the transmission of a scheduled TT frame. Thus,
shuffling improves the bandwidth utilization for BE and RC
traffic, while increasing the jitter for the TT frames.

We define the required bandwidth by BE traffic on a
dataflow link \( l_i \), \( BW_{BE_{\text{req}}}(l_i) \) as the sum of the required bandwidth
by all the BE messages transmitted over \( l_i \).

Similarly, \( BW_{TT_{\text{req}}}(l_i) \) and \( BW_{RC_{\text{req}}}(l_i) \) define the necessary band-
width for TT and RC frames on \( l_i \). Thus, we define the available bandwidth for BE traffic on a dataflow link \( l_i \) as
where \( BW^0 \) is the initial bandwidth of the dataflow link.
III. PROBLEM FORMULATION

Our problem can be formulated as follows: given the network topology \( G \), the set of messages \( M \) (including the size and period/rate for all messages, and the deadline for the TT and RC messages), we are interested to optimize the network implementation \( Y \) such that the TT and RC frames are schedulable, and the bandwidth for the BE frames is maximized.

We exemplify our problem using the topology presented in Fig. 1. The solution presented in Fig. 2a was obtained with DOTTS, which optimizes the network implementation considering only TT and RC messages, and ignoring the BE traffic. Moreover, in DOTTS we consider that the BE frames are routed along the shortest route. Thus, in this configuration dataflow link \([NS_1, NS_2]\) does not have sufficient bandwidth for all the BE frames. However, taking into account the BE traffic when optimizing the network implementation leads to solutions which provide more bandwidth for the BE traffic, while at the same time guaranteeing the timing constraints of the TT and RC traffic, see Fig. 2b. The solution in Fig. 2b was obtained by rerouting the RC frame \( RC_3 \) and the BE frames \( BE_1 \) and \( BE_3 \) from the virtual link \([ES_2, NS_1]\), \([NS_1, NS_2]\), \([NS_2, ES_2]\) considered in Fig. 2a, through the virtual link composed of dataflow links \([ES_2, NS_1]\), \([NS_1, NS_2]\), \([NS_2, ES_2]\).

The increase in bandwidth available for BE messages can also be obtained, in general, by rescheduling TT frames, rerouting TT and RC frames or by packing / fragmenting.

IV. DESIGN OPTIMIZATION OF TTEthernet-BASED SYSTEMS

Next, we will summarize the DOTTS strategy, presented in detail in [18]. DOTTS takes as input the topology of the network \( G \) and the set of TT and RC messages \( M^{TT} \cup M^{RC} \) (including the size, period/rate and deadline), and returns (i) the fragmenting of messages \( \Phi_m \) and packing in frames \( K \), (ii) the assignment of frames to virtual links \( M_F \), (iii) the routing \( R_{VL} \) of virtual links, (iv) the bandwidth \( B \) for each RC virtual link and (v) the schedules \( S \) for the TT frames. We denote with \( Y \) the tuple \( < \Phi_m, K, M_F, R_{VL}, B, S > \) obtained by running the DOTTS algorithm.

DOTTS is a Tabu Search-based metaheuristic, which searches for that solution that minimizes the cost function:

\[
Cost = w_{TT} \times \delta_{TT} + w_{RC} \times \delta_{RC}
\]

where \( \delta_{TT} \) and \( \delta_{RC} \) are the degree of schedulability for the TT and RC frames, respectively. These are summed together into a single value using the weights \( w_{TT} \) and \( w_{RC} \), given by the designer. In case a frame is not schedulable, its corresponding weight is a very big number, i.e., a “penalty” value. This allows us to explore unschedulable frames in the hope of driving the search towards a feasible region. Once the TT frames are schedulable we set the weight \( w_{TT} \) to zero, since we are interested to minimize the end-to-end delays for the RC frames. The degree of schedulability is calculated as:

\[
\delta_{TT/RC} = \begin{cases} 
  c_1 = \sum \max(0, R_f - f_i \cdot \text{deadline}) & \text{if } c_1 > 0 \\
  c_2 = \sum (R_f - f_i \cdot \text{deadline}) & \text{if } c_1 = 0
\end{cases}
\]

If at least one frame is not schedulable, there exists one \( R_f, \) i.e., the worst-case end-to-end delay (WCD) of \( f_i \), greater than the deadline \( f_i \cdot \text{deadline} \), and therefore the term \( c_1 \) will be positive. We have discussed in [17] how the WCD of a frame is calculated. However if all the frames are schedulable, this means that each \( R_f \) is smaller than \( f_i \cdot \text{deadline} \), and the term \( c_1 = 0 \). In this case, we use \( c_2 \) as the degree of schedulability, since it can distinguish between two schedulable solutions.

Tabu Search explores the design space by using design transformations (or “moves”) applied to the current solution in order to generate neighboring solutions. DOTTS has three classes of moves: (1) routing moves applied to virtual links, (2) packing moves applied to messages and (3) scheduling moves applied to the TT frames.

V. OPTIMIZATION FOR BE-TRAFFIC

DOTTS is a flexible framework that can be used to optimize different aspects of a TTEthernet network. We will show next how to modify DOTTS to obtain schedulable implementations that maximize the available bandwidth for the BE traffic. The first step is to modify the cost function used by DOTTS from Eq. 4 to the following:

\[
\text{Cost} = \begin{cases} 
  c_1 = \sum \max(0, R_f - f_i \cdot \text{deadline}) & c_1 > 0, f_i \in M^{TT} \cup M^{RC} \\
  c_2 = \sum \max(0, BW_{BE}(l_j) - BW_{avail}(l_j)) & c_1 = 0 \text{ and } c_2 > 0 \\
  c_3 = \sum (BW_{BE}(l_j) - BW_{avail}(l_j)) & c_1 = 0 \text{ and } c_2 = 0
\end{cases}
\]

Once all the TT and RC frames are schedulable, i.e., each \( R_f \) is smaller than the deadline \( f_i \cdot \text{deadline} \), and as such, \( c_1 = 0 \), DOTTS will search for a solution that satisfies the bandwidth requirements of the BE frames. If at least one dataflow link \( l_j \in DDL \) exists, where the \( BW_{BE}(l_j) \) (Eq. 2) bandwidth required by the BE frames routed via \( l_j \) is larger than the available bandwidth \( BW_{avail}(l_j) \) (Eq. 3), the BE traffic has insufficient bandwidth, i.e., \( c_2 > 0 \), and DOTTS will use \( c_2 \) as the value of the cost function. In case all the dataflow links have sufficient available bandwidth to satisfy the BE traffic, \( c_2 = 0 \), and the value of the cost function will be \( c_3 \). This cost function will drive DOTTS towards solutions that satisfy the schedulability of TT and RC traffic and that maximize the available bandwidth for BE frames.

The second extension of DOTTS is concerned with new moves. In DOTTS, the reroute move focuses only on TT and RC frames, and the choice of new VLs was made randomly.
We introduce a reroute move for BE traffic, which selects BE frames from the most “congested” (i.e., highest bandwidth utilization) dataflow link and reroutes them via the least used links. We also introduce a packing move for BE messages, which selects the small BE messages routed on the same VL, and packs them together into a frame, thus reducing the incurred protocol overhead. We denote with DOTTTS+ the application approach presented in this section.

VI. EXPERIMENTAL EVALUATION

We evaluated the new DOTTTS+ strategy using the real-life case study Orion Crew Exploration Vehicle (CEV), derived from [8]. The topology for the case study is shown in Fig. 3. The Orion CEV case study has 31 ESes and 13 NSes, connected by dataflow link transmitting at 100 Mbps, and 187 TT and RC messages, with parameters generated based on the messages presented in [8]. At the network level, the TT and RC messages require a little over 50 Mbps. For BE traffic, we added 123 messages with bandwidth requirements that vary between 96 kbps (the average internet radio stream) and 10 Mbps (the read/write speed of DVD at 1x speed). The BE messages require 351 Mbps of bandwidth at the network level. Overall, these messages resulted in 52,815 frame instances transmitted over the network. DOTTTS is not able to accommodate BE traffic in the network resulting in dataflow links which have zero bandwidth for BE frames. However, we ran our DOTTTS+ optimization for 90 minutes, we managed to obtain a solution which satisfies the deadlines of the TT and RC messages, and provides 73.47% of the bandwidth required by BE messages. The BE frames have enough bandwidth for transmission on 88 out of the 94 dataflow links.

We were also interested to determine the impact of the timely blocked integration policy, compared to shuffling, on the BE messages. Thus, the unusable bandwidth on a dataflow link due to the timely block intervals ranges from 0 Mbps (no TT frames transmitted on that link) to 7.36 Mbps. At the whole network level, the timely block intervals take up 198 Mbps of bandwidth. On the 6 dataflow links the BE frames do not have enough bandwidth for transmission, 33.65 Mbps of bandwidth is taken by the timely block intervals. In case the time constraints for TT messages are less stringent, and the TT traffic can tolerate a higher jitter, the timely blocked intervals would be eliminated by implementing the shuffling policy, thus satisfying 83.06% of the required bandwidth for BE traffic.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a design optimization strategy to take into account BE traffic in TTEthernet-based systems. The proposed strategy, named DOTTTS+, optimizes the network implementation such that the TT and RC messages are schedulable, and the available bandwidth for BE traffic is maximized. We have evaluated the DOTTTS+ strategy using the Orion CEV real-life test case, and we have also shown the impact of timely traffic integration policy on the BE traffic.

DOTTTS+ is based on a Tabu Search meta-heuristic. As future work, we will implement a heuristic-based approach, which would considerably reduce the necessary time to obtain a solution. Moreover, in this paper we modeled BE messages as aperiodic, with a minimum inter-arrival time. However, for many non real-time applications it is not possible to derive such parameters. Thus, another challenge is to improve the BE traffic model. Finally, we will evaluate DOTTTS+ on several real-life and synthetic benchmarks.

Fig. 3: Network topology of the Orion CEV, derived from [8]

REFERENCES