SCIENCE CHINA
Information Sciences

• LETTER •

January 2018, Vol. 61 019102:1–019102:3
https://doi.org/10.1007/s11432-017-9121-6

An imporosity message scheduling based on modified genetic algorithm for time-triggered Ethernet

Yingjing ZHANG, Feng HE* , Guangshan LU & Huagang XIONG

School of Electronic and Information Engineering, Beihang University, Beijing 100191, *China*

Received 15 March 2017/Accepted 25 May 2017/Published online 4 December 2017

Citation Zhang Y J, He F, Lu G S, et al. An imporosity message scheduling based on modified genetic algorithm for time-triggered Ethernet. Sci China Inf Sci, 2018, 61(1): 019102, https://doi.org/10.1007/s11432-017-9121-6

Dear editor,

In time-triggered Ethernet (TTEthernet), there are three kinds of messages which are time-triggered (TT), rate-constrained (RC) and best-effort (BE) messages. In general, TTEthernet scheduling can be considered to be a multi-resources scheduling, which is proven to be a non-deterministic polynomial (NP) complete problem [1]. In [2] a priori porosity schedule is proposed based on satisfiability modulo theories (SMT). In this method, TT packets will be sent back-to-back and all of the RC packets will be arranged after TT packets at the last part of each cluster cycle (CC). However, long latencies of RC packets will occur since all of RC packets are arranged in the back of CC. To solve this problem, a posteriori porosity schedule is proposed [3]. Unfortunately, the suboptimal schedule is still not obtained despite its improvement in the latencies of RC packets. In this article, an imporosity scheduling algorithm is proposed based on a modified GA. Different from the porosity schedules, the imporosity scheduling algorithm can assign TT and RC packets together. Although RC packets do not have any exact arriving time, their arriving windows can be known. Subsequently, end-to-end latencies of RC packets can be calculated with their scheduling and arrival time. The purpose of this article is to minimize end-to-end latencies by appropriately designing the scheduling sequences in the whole network. In existing literature, genetic algorithms (GA) have been widely used for scheduling in computing areas. In [4], a suboptimal scheduler is proposed for real-time systems by using GA. In [5], soft real-time message scheduling problem can be solved by a multi-objective hybrid GA. Moreover, optimal task scheduling in cloud service can also be solved by using GA [6]. Similarly, a modified GA is proposed for our imporosity schedule in this article. According to this method, determination of both TT and RC communications is considered from senders to receivers with the aid of partition in ARINC 653. As a result, the suboptimal schedule can be obtained where shorter end-to-end latencies of RC packets can be achieved.

There are different messages on physical communication links. Based on the TT scheduling tables generated in an off-line method, TT messages are required to be sent within predefined time windows. The schedule tables containing dispatch points of messages should be stored in each end systems. As for TT_i messages, their information should include the period TT_i.period, the length TT_i.length and the scheduled moment TT_i.offset. Compared to TT, RC messages are less strict for real-time requirements. Meanwhile, RC messages may be dispatched by different communication controllers at the same point, where RC messages may queue up in the switches as a consequence. For RC messages, the minimum inter-frame gap BAG denoted by RC_j.BAG and

* Corresponding author (email: robinleo@buaa.edu.cn)

the length represented as RC_j.length are included. Moreover, the partition model is proposed in the ARINC 653 standard [7], where the dispatched moments of RC messages are constrained by partition windows for an integrated avionics system. Therefore, the arrival time RC_j.arrival can be assumed to be known.

The procedure of scheduling algorithm based on modified GA method can be summarized as follows, where five steps are included.

Step 1. Initialization. On each physical link, the arrangement of packets is considered to be a chromosome and each packet is considered as the gene on the chromosomes. The initial population can be formed by w (w is an even number) chromosomes. It should be noted that the initial population cannot be simply obtained with randomly generated chromosomes, since the chromosomes should satisfy the periodic scheduling constraints. For example, there are several packets $\mathrm{TT}_1(1), \mathrm{TT}_1(2), \mathrm{RC}_1(1)$ to be scheduled where $\mathrm{TT}_1(1)$ and $\mathrm{TT}_1(2)$ are the 1st and 2nd period packets of TT_1 respectively. Since $\mathrm{TT}_1(2)$ should be sent after $\mathrm{TT}_1(1)$, there are only three acceptable sequences, i.e., $\mathrm{TT}_1(1), \mathrm{TT}_1(2), \mathrm{RC}_1(1)$, $\mathrm{TT}_1(1), \mathrm{RC}_1(1), \mathrm{TT}_1(2)$ and $\mathrm{RC}_1(1), \mathrm{TT}_1(1), \mathrm{TT}_1(2)$. Therefore, it is not suitable to use binary encoding in traditional GA for generating random chromosomes. In contrast, packets are directly used as genes. Furthermore, the sequences can be generated based on FIFO policy. In particular, early-coming messages will be sent early based on FIFO policy, where the constraints of periodic scheduling can be satisfied. Compared to the randomly selected sequences, FIFO policy can be used to shorten end-to-end latencies of packets and accelerate the convergence of the algorithm.

Step 2. Calculation of individual's fitness values. It is a critical factor to calculate the fitness values of individuals since it will affect the optimal individuals seeking process and evolutionary performance of GA.

The chromosome in whole network is denoted by σ_ℓ. There are n RC and m TT messages. The objective function value of the sequence σ_ℓ is defined as follows:

$$\mathrm{z}\left(\sigma_\ell\right) = \sum_{j=1}^{n} h_j, \tag{1}$$

where h_j is the end-to-end latency of message RC_j. More specifically, h_j can be calculated depending on the sequence σ_l and the arrival moment RC_j.arrival.

However, there exists a trade-off between diversity of population and local convergence for the objective function value. Therefore, a fitness scaling becomes necessary for solving this problem [8]. In this article, exponent function is used for fitness scaling,

$$Z_\ell = \mathrm{e}^{-z(\sigma_\ell)/\varphi}, \tag{2}$$

where Z_ℓ is the fitness after scaling and φ is the scaling parameter obtained by calculating the mean of the maximum $z(\sigma)_{\max}$ and minimum $z(\sigma)_{\min}$.

In this article, the store-and-forward time of each switch is simply ignored. The minimum objective function value exists if the packets are scheduled once they arrive, where the value should be zero. The maximum objective function value can be obtained by network calculus [9].

The maximum length of RC_j is $\mathrm{RC}_j.\mathrm{length}_{\max}$ and the arrival curve of RC_j message can be given by

$$\alpha(t)_{\mathrm{RC}_j} = \frac{\mathrm{RC}_j.\mathrm{length}_{\max}}{\mathrm{BAG}_j} \times t + \mathrm{RC}_j.\mathrm{length}_{\max}. \tag{3}$$

The service curve for RC_j message provided by the switch can be represented

$$\beta(t)_{\mathrm{TT}_i} = C^* \times [t - \mathrm{TT}_i.\mathrm{length}_{\max}]^+, \tag{4}$$

where C^* is the rate of RC messages, T_{RC} is the waiting time of RC messages impacted by TT messages, C is the rate of physical links and $\mathrm{TT}_i.\mathrm{length}_{\max}$ is the maximum length of TT_i. Moreover, C^* and T_{RC} can be described as follows:

$$C^* = C - \sum_{1\leqslant i\leqslant m} \frac{\mathrm{TT}_i.\mathrm{length}_{\max}}{\mathrm{TT}_i.\mathrm{period}}, \tag{5}$$

$$T_{\mathrm{RC}} = \frac{\sum_{1\leqslant i\leqslant m} \frac{\mathrm{TT}_i.\mathrm{length}_{\max}}{\mathrm{TT}_i.\mathrm{period}}}{C}. \tag{6}$$

Once the messages are sent after a switch, the new arrival curve $\alpha'(t)$ can be obtained by the deconvolution of the arrival curve and service curve. The service curve of two switches can be acquired by the convolution of the two service curves.

The maximum end-to-end latency $z(\sigma)_{\max}$ is defined as the horizontal deviations between the final arrival curves and service curves.

Step 3. Crossover. Due to randomness in genetic manipulations including crossover and mutation, the best adaptive individuals are possible to be destroyed, which may have some impacts on operating efficiency and convergence of genetic algorithms. To avoid this situation, an elitist strategy is employed before crossover procedure. In this strategy, the worst individual is replaced by the best individual before genetic manipulations to keep the best individual in the next generation.

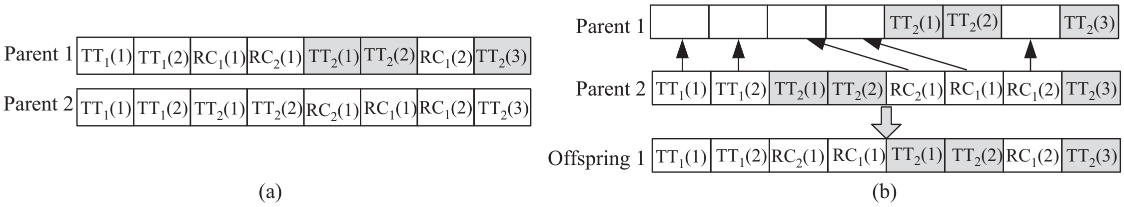

Figure 1 The process of position based crossover. (a) Marking in Parent 1; (b) producing Offspring 1.

Position based crossover (PBC) can be performed as follows. Firstly, the messages are chosen with a constant mutation rate P_c. Secondly, the selected message positions can be obtained, which are marked in Parent 1 as shown in Figure 1(a). Afterwards, the unmarked messages in Parent 2 are placed into unmarked positions in Parent 1, where the Offspring 1 is produced as shown in Figure 1(b). Finally, the roles of Parents 1 and 2 are reversed and Offspring 2 can be obtained.

Step 4. Mutation. The individuals are selected with a constant mutation rate P_m. Afterwards, the selected individual is exchanged with the next one in the sequence. It should be noted that the constraints of periodic scheduling shall be satisfied. Otherwise, the selected individual exchanges its position with the message after the following one instead.

Step 5. Termination. In Step 1, the reserved individual reproduces itself, where the individuals can be obtained by adding the new $w-2$ individuals. Once the number of generation reaches the maximum value, the algorithm will be terminated and will search for an optimal individual; otherwise, go to Step 2.

Experiment simulations in this article are shown in Appendix A.

Conclusion. In this article, an imporosity schedule is proposed, which is based on modified GA and the partition model for integrated avionics system. In contrast to porosity schedule based on SMT algorithm, the proposed algorithm can not only provide a suboptimum schedule, but also can satisfy the periodic constraints in TTEthernet. The major advantage of the method is that end-to-end latencies of RC messages can be substantially decreased. More importantly, with the increase in packets load, the improvement can demonstrate an increasing trend. Although the synthesis time of the proposed algorithm could be longer than that of posteriori porosity schedule, it is fast enough for static off-line scheduling practically, where requirements of industrial development process can be satisfied.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61301086, 20131951027) and China Scholarship Council.

Supporting information Experiment simulations (Appendix A). The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Craciunas S S, Oliver R S, Ecker V. Optimal static scheduling of real-time tasks on distributed time-triggered networked systems. In: Proceedings of IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, 2014. 1–8
2 Craciunas S S, Oliver R S. SMT-based task and network-level static schedule generation for time-triggered networked systems. In: Proceedings of the 22th International Conference on Real-Time Networks and Systems, Versaille, 2014. 45–54
3 Steiner W. Synthesis of static communication schedules for mixed-criticality systems. In: Proceedings of the 14th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops. Washington: IEEE Computer Society, 2011. 11–18
4 Miryani M R, Naghibzadeh M. Hard real-time multi-objective scheduling in heterogeneous systems using genetic algorithms. In: Proceedings of the 14th International CSI Computer Conference (CSICC), Tehran, 2009. 437–445
5 Yoo M, Gen M. Scheduling algorithm for real-time tasks using multi-objective hybrid genetic algorithm in heterogeneous multiprocessors system. J Comput Oper Res, 2007, 34: 3084–3098
6 Gui H Y, Li Y, Liu X F. Cloud service reliability modelling and optimal task scheduling. IET Commun, 2017, 11: 161–167
7 Airlines Electronic Engineering Committee (AEEC). Avionics Application Software Standard Interface-ARINC Specification 653-Part 2, 2007
8 Marcus H, Shane L. Fitness uniform optimzation. IEEE Trans Evolution Comput, 2006, 10: 568–589
9 Markus F, Amr R. A guide to the stochastic network calculus. IEEE Commun Surv Tut, 2015, 17: 92–105