SCIENCE CHINA
Information Sciences

• **Supplementary File** •

An imporosity message scheduling based on modified genetic algorithm for Time-Triggered Ethernet

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

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

Appendix A Experiment simulations

The topology and packets distribution are shown in Figure A1. The transmission rate of the physical link is $C = 100Mbps$ and the lengths of packets are uniformly selected from 64 to 1518 bytes. The periods of TT flows and BAGs of RC flows are given in Table A1. Based on the partition model of the integrated avionics system, the dispatched moments of RC messages in cluster cycle (CC) can be constrained by partition windows as shown in Figure A2.

In this experiment, 50 Monte Carlo simulations performed to calculate and compare the average end-to-end latencies of each RC message as shown in Figure A3. The proposed GA-based schedule can select a population size of 100, a crossover rate of 1.0 and a mutation rate of 0.01. In Figure A3, the simulated results on the average end-to-end latency of each RC message are shown, which are obtained by posteriori porosity schedule policy and modified GA schedule, respectively. The results obtained by porosity schedule policy and modified GA-based algorithm are represented by red dots and blue dots, respectively. It can be observed that the average end-to-end latency of each RC message obtained by GA-based scheduling is less than that obtained by the posteriori porosity scheduling.

In Figure A4, the relationship between the number of generations and fitness values of individuals in modified GA can be revealed. The fitness values of individuals can increase with the increase in generation number utile convergence. Meanwhile, it is apparent that the modified GA method can converges within 50 iterations, which indicates easy convergence of the modified GA method. Although the proposed GA can facilitate fast convergence, it is possible that the obtained result is local optima rather than the global optima. Therefore, the algorithm terminates once reaching a satisfactory suboptimal solution or producing a predefined number of generations. It should be noted that the latter policy is adopted in this paper. Furthermore, simulations are performed to compare the fitness values with 1000 generations and 50 generations respectively. After 50 generations, the fitness values can be 0.918, which is very close to the value 0.923 after 5000 iterations. Hence, it is reasonable to consider that our method can easily converge to a satisfactory solution.

In the following, we will discuss schedule synthesis times of proposed imporosity scheduling algorithm in comparison with that of the posteriori porosity schedule. In particular, both algorithms are implemented on an Intel Core 2 Duo 2.8 GHz PC. In Figure A5, the synthesis times versus the number of messages are represented from one hundred to seven hundred. Note that seven hundred messages can take up 95% bandwidth. Each dots represents the synthesis time of randomly generated messages.

It is not surprising that the synthesis times of our algorithm are higher than that of posteriori porosity schedule. The reason is that a suboptimal schedule is obtained in the proposed algorithm rather than a feasible solution. In fact, we are mostly concerned with static off-line scheduling problem, where synthesis times are secondary to the end-to-end latencies. In addition, the longest schedule synthesis time is discussed for our algorithm. Since the CC is $128ms$ and the average length of messages is $64\mu s$ in practice, the number of messages can reach 2000 in the CC. Therefore, the genes on chromosomes in GA should be no more than 2000. The longest schedule synthesis time is half an hour simulation. Notably, the length of the time is short enough to meet the requirements in industrial development processes.

Moreover, scenarios with different message loads on the network are simulated. On account of posteriori porosity schedule, the time slices for RC messages are decentralized, which will increase the possibility of resources wastage. The reason is that SMT policy only seeks one feasible schedule for TTEthernet communication. If resources among TT messages are not abundant for RC messages in each time slice, it is required to be aborted. In contrast to posteriori porosity schedule, the proposed modified GA-based schedule can obtain a suboptimum schedule by pre-planning procedure. The advantage of modified GA-based schedule becomes more prominent with the increase in packet load of the whole network, which can be observed in Figure A6.

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

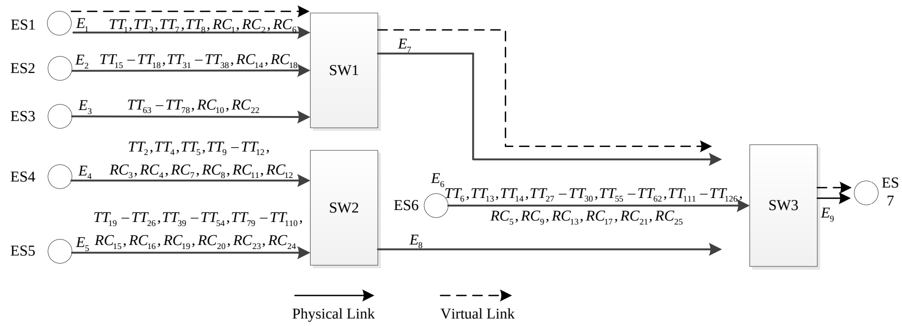

Figure A1 An example of TTEthernet network architecture.

Table A1 Configurations of the packets

TT Flows	Periods	RC Flows	BAGs (μs)
$TT_1 - TT_2$	$2ms$	$RC_1 - RC_5$	$2ms$
$TT_3 - TT_6$	$4ms$	$RC_6 - RC_9$	$4ms$
$TT_7 - TT_{14}$	$8ms$	$RC_{10} - RC_{13}$	$8ms$
$TT_{15} - TT_{30}$	$16ms$	$RC_{14} - RC_{17}$	$16ms$
$TT_{31} - TT_{62}$	$32ms$	$RC_{18} - RC_{21}$	$32ms$
$TT_{63} - TT_{126}$	$64ms$	$RC_{22} - RC_{25}$	$64ms$

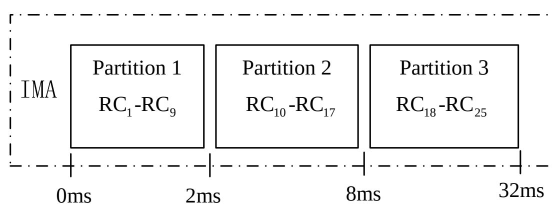

Figure A2 The partition windows for *RC* messages.

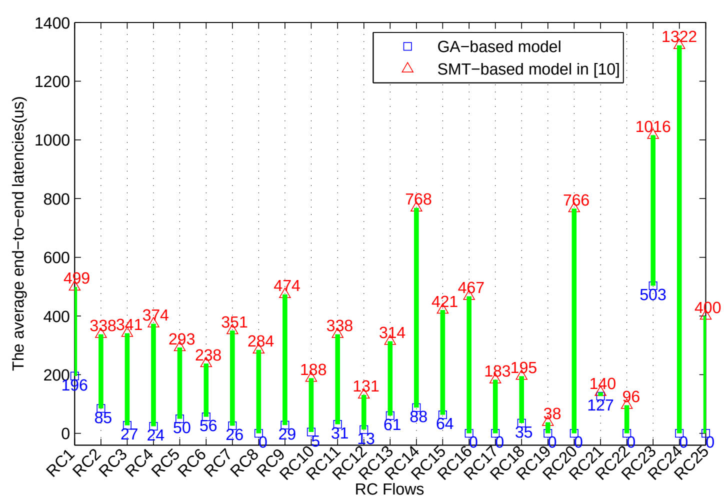

Figure A3 The average end-to-end latencies of *RC* packets.

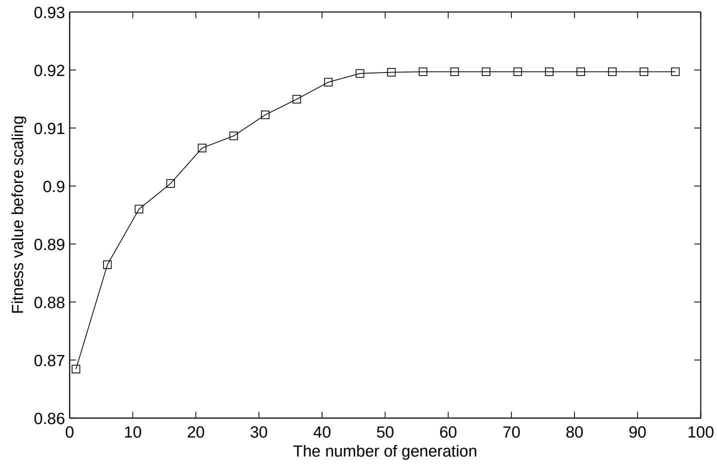

Figure A4 The relationship between iteration number of GA and fitness values of individuals.

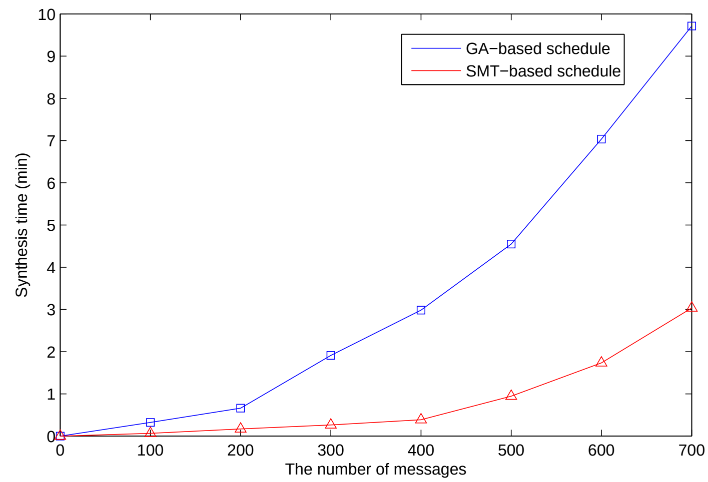

Figure A5 The relationship between iteration number of GA and fitness values of individuals.

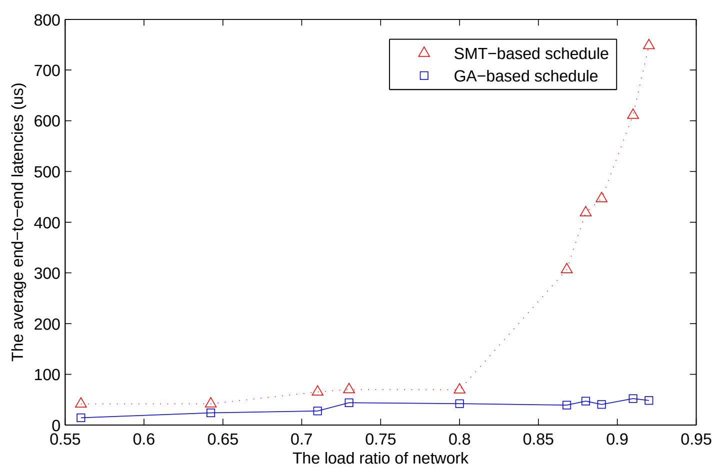

Figure A6 The average end-to-end latencies with different load of network.