Research on Time Triggered Ethernet Scheduling Planning Method

Aodi Wei*, Guoqun Zhang² and Tao Zhang¹

¹School of software, Northwestern Polytechnical University, Xi'an, China
²Shanghai Mechanical & Electrical Engineering Research Institute, Shanghai, China

*Email address: 892363018@qq.com

Abstract. Time-triggered Ethernet adds a time-triggered traffic type based on traditional event-triggered best-effort traffic, which extends data traffic into three communication formats, namely time-triggered (TT) messages and rate-constrained (RC) messages and the best-effort (BE) messages. Aiming at the time-triggered Ethernet message scheduling planning problem, this paper first introduces the principle of the SMT solver and the idea of the solver algorithm, and designs constraints for the message scheduling problem. Based on this, the paper introduces a load balancing strategy and a step-by-step iterative conflict backtracking method, designs and completes a time-triggered Ethernet message scheduling algorithm, and then completes scheduling planning for time-triggered traffic and optimization of scheduling results. For the planning results, the paper proposes verification indicators for network load balancing and algorithm solution efficiency, designs a test set for algorithm verification, completes algorithm analysis and verification based on the test set solution results, and determines the algorithm optimization effect.

1. Introduction

Time-triggered Ethernet (TTE) is a solution to guarantee the real-time performance of Ethernet [1]. The time-triggered messages are sent strictly according to the schedule, and the transmission of other types of messages is stopped during the transmission, thereby avoiding link contention and message retransmission, and ensuring the real-time transmission and reliability requirements.

Since the scheduling schedule of the time-triggered messages needs to be generated before the network scheduling starts, the quality of the scheduling schedule also determines the scheduling efficiency [2]. Thus, the core of research on time-triggered Ethernet is the study of network scheduling planning algorithms. At present, the mainstream research on time-triggered Ethernet scheduling problems has the following ideas:

One idea is that the scheduling problem can be expressed as a solution to the system scheduling constraint. The existing SMT solver can be used to obtain a feasible solution that satisfies the condition by a given constraint set [3]. In addition, Dvorak J et al propose to treat the problem as a resource constraint project scheduling problem, which was solved by extending the SCIP solver [4]. Another idea is to solve iteratively by heuristic algorithm and sub-heuristic algorithm. For example, Domitian Tamas-Selicean [5] proposed a scheduling schedule design based on Tabu Search Algorithm. Another way to refer to traditional real-time system scheduling algorithm, such as a TT-RMS-based schedule generation algorithm, proposed by Beihang University Xu Xiaofei et al [6], and improved by Air Force Engineering University Zhang Chao [7].
2. Time-triggered Ethernet Scheduling Algorithm

2.1. Time Triggered Ethernet Scheduling Planning Constrains

The TT message scheduling problem can be seen as scheduling satisfiability problem. The solution is to express the constraint that the message scheduling needs to be satisfied in the form of SMT formula, constraint solving for message scheduling moments with SMT solver. Therefore, how the TT message scheduling constraint is standardized into the form of the SMT formula is an important issue for research. The TTE Ethernet topology consists of nodes, switches, and physical links. The physical link uses full-duplex communication. This paper uses the undirected graph $G (V, E)$ to describe the network topology. The set $V = \{v_1, v_2, v_3 \cdots v_n\}$ is the switch and host node in the topology, and set $E$ is the physical link connecting the host and the switch. In the planning, this paper denotes the undirected data link with $(x, y)$, and the undirected physics of the node $v_i$ to the node $v_j$ by $(v_1, v_2)$ in the network $G$. Thus, the set $E = \{(v_1, v_2), (v_3, v_2)\}$. The directional data stream link directly connected from node $v_i$ to node $v_j$ can be expressed as $[v_i, v_j]$, and if $L$ is its data stream link set, then $L = \{([v_1, v_2], [v_3, v_2])\}$.

Time-triggered Ethernet introduces the concept of virtual link (Virtual Link VL) to the forwarding path of messages. The switch and the node complete the message forwarding according to the time window specified by the VL. On this basis, the ordered data stream link set VL of the message from the source node to the destination node can be expressed as:

$$p = \left[ [v_i, v_k], \cdots, [v_r, v_j] \right]$$ (1)

Where $v_i$ is the source node, $v_j$ is the destination node, and $v_k$ and $v_r$ are intermediate nodes that pass sequentially, $p$ is the VL of the message from $v_i$ via $v_k$, $v_r$, etc. to $v_j$. In the TTE network, in the link specified by VL, the message is forwarded in the form of a frame. Therefore, the definition set $F$ represents the frame transmitted in the message link, and $f_i$ represents the message frame transmitted in the specified VL link $v_l$ in the set $F$. The message scheduling problem becomes a frame scheduling problem in VL units.

For frame scheduling, when frames are forwarded on a designated link, we mainly focus on three attributes of the frame: frame length, frame period, and frame transmission offset. Data frames in a TTE network can be represented by the following triples:

$$f_i = \{f_i.\text{length}, f_i.\text{period}, f_i.\text{offset} \}$$ (2)

$f_i.\text{period}$ indicates the transmission period of the frame; $f_i.\text{length}$ indicates the length of the message described by the time, which is related to the length of the message frame byte and the network port rate; $f_i.\text{offset}$ is the transmission offset of the message, and their units are all ms or $\mu$s.

In this paper, the forwarding process of the frame on the path is decomposed, and each forwarding process of the frame is divided by the data stream link in the VL set $p$. According to the TTE Ethernet standard, the TT frame $f_i$ on the data stream link $[v_k, v_l]$ can be represented by the following triples:

$$f_i^{[v_k, v_l]} = \{f_i^{[v_k, v_l]}.\text{length}, f_i^{[v_k, v_l]}.\text{period}, f_i^{[v_k, v_l]}.\text{offset} \}$$ (3)

$f_i^{[v_k, v_l]}.\text{length}$ and $f_i^{[v_k, v_l]}.\text{period}$ are determined by the imported message list, $f_i^{[v_k, v_l]}.\text{offset}$ represents the frame $f_i$’s transmission offset at the link $[v_k, v_l]$, that is, the transmission time of $f_i$ at the terminal $v_k$, which is the target value of the desired solution.

According to the TTE scheduling specification, in frame scheduling, the frame needs to meet the following constraints: the scheduling of the frame should satisfy the actual situation, that is, the frame offset should be non-negative; when the frame is sent from the source node, the offset cannot exceed the period of the frame; In the same physical connection, only one frame can be sent at the same time; in the process of forwarding frames, the forwarding sequence of the virtual link cannot be violated; the frame must meet the hardware requirements of the TTE switch; In the case of multiple channels, it should be ensured that the node sends a single frame to different links at the same time; in the case of
broadcast, the switch should send frames to different data links at the same time. Formally, the constraint specification to the SMT formula can be expressed as follows:

1. Basic constraint
   - The basic constraint ensures that the scheduling of the frame meets the actual background requirements, and the frame offset can only be non-negative, otherwise it cannot be sent. The constraint is as shown in formula (4):
     \[ \forall f_i \in F \ f_i.offset \geq 0 \]  
     \[ (4) \]

2. Single period constraint
   - When the frame is sent by the source node, it should be guaranteed to be sent within the first frame period. It should be noted that this constraint is not added during the frame forwarding process. And is supplemented by other constraints in the intermediate nodes and destination nodes of the frame.
     \[ \forall f_i \in F \ f_i^{first(f_i)}.offset < f_i.period \]  
     \[ (5) \]
   - Where \( first(f_i) \) represents the source node of the frame \( f_i \), then \( f_i^{first(f_i)} \) represents the data frame sent by the message from the source node.

3. No conflict constraint
   - This constraint addresses link contention issues that exist in data transmission. This constraint stipulates that in the data link, the message should be sent mutually exclusive, and the message frame can only be sent after the other frames on the same link are sent. Therefore, the constraint achieves collision-free constraints by limiting the transmission mutex between two frames in the data link. The constraint is as shown in formula (6).
     \[ \forall [V_k, V_l] \in L; \forall f_i, f_j \in F; 
     \forall a \in \left[ 0 \ldots \left( \frac{LCM}{f_i.period} - 1 \right) \right]; \forall b \in \left[ 0 \ldots \left( \frac{LCM}{f_j.period} - 1 \right) \right]; 
     \left( f_i \neq f_j \right) \land \exists f_i^{[V_k, V_l]} \land \exists f_j^{[V_k, V_l]} \implies f_i^{[V_k, V_l]} offset + (a \times f_i.period) \neq f_j^{[V_k, V_l]} offset + (b \times f_j.period) \]  
     \[ (6) \]

4. Path related constraint
   - The path correlation constraint describes the sequence requirement of frame forwarding in VL path. The forwarding path of frame must be executed in strict accordance with the sequence of nodes in the path. The scheduling of frames on subsequent data stream links will be executed after the previous data stream link is received. This can be constrained by restricting the scheduling order of its adjacent two hops. For example, when the switch carries out frame forwarding, the frame should be executed in the order of receive, cache and forward. The constraint is shown in formula (7):
     \[ \forall v_l \in VL; \forall p \in vl; \forall [v_1, v_m], [v_m, v_n] \in p; 
     f_i^{[v_1, v_m]} - f_i^{[v_m, v_n]} \geq max_{hop.Sequential} \]  
     \[ (7) \]

5. Switch is bounded constraint
   - The switch bounded constraint stipulates that the scheduling time of frames must be within the range allowed by the switch hardware. This constraint stipulates the upper limit of message storage time in the switch, and the constraint is shown in formula (8):
     \[ \forall v_l \in VL; \forall p \in vl; \forall [v_1, v_m], [v_m, v_n] \in p; 
     f_i^{[v_1, v_m]} - f_i^{[v_m, v_n]} \leq mem.cons \]  
     \[ (8) \]
   - Where \( mem._cons \) is the memory-related parameter of the switch.

6. Simultaneous dispatch constraint of switch
In broadcast communication, the switch may need to forward the same frame on multiple output links. In this case, it is necessary to guarantee the simultaneity of such broadcast, and then it is necessary to restrict the synchronous transmission of frames. The constraint is shown in formula (9):

\[ \forall v \in \mathcal{V}; \forall p_a, p_b \in v; \forall [v_j, v_k] \in p_a; \forall [v_j, v_l] \in p_b \]

\[ f_{i \left[v_j,v_k\right]} \_offset = f_{i \left[v_j,v_l\right]} \_offset \]

The algorithm divides the message cluster into smaller clusters and solves one cluster at a time. When the solution fails, additional constraints are imposed on the solution to make it more optimized. These constraints are not required to be met. When the solver determines a conflict, the constraints can be relaxed or deleted.

The additional soft constraint requires that the frame meets the same data stream link, and each frame interval should be greater than a fixed value. The constraint is as shown in formula (10):

\[ \forall [v_k, v_l] \in L; \forall f_i, f_j \in F; \]

\[ \forall a \in \left[0, \left(\frac{\text{LCM}_{f_i, \_period}}{f_i, \_period} - 1\right)\right]; \forall b \in \left[0, \left(\frac{\text{LCM}_{f_j, \_period}}{f_j, \_period} - 1\right)\right]; \]

\[ \left| f_{i \left[v_k,v_l\right]} \_offset + (a \times f_i, \_period) - (f_{j \left[v_k,v_l\right]} \_offset + (b \times f_j, \_period)) \right| \geq \text{Min\_interval} \]

Min\_interval is the minimum interval between two frames. Its value is related to the number of frames to be sent in the data stream link and the length of the cluster cycle. The relationship between the three is shown in formula (11):

\[ \forall [v_k, v_l] \in L; \ rac{\text{LCM}}{\sum_{i=0}^{n} f_{i \left[v_k,v_l\right]} \_length} \geq \text{Min\_interval} \]

Ideally, Min\_interval is equal to the ratio of the number of frames to the length of the cluster period. In actual planning, Min\_interval will be less than the ideal interval. In the case that the solver cannot solve the soft constraints, the optimization scheme by adjusting Min\_interval.

The algorithm uses the greedy strategy to adjust the frame interval Min\_interval. When determining the initial frame interval, the algorithm uses the ideal interval multiplied by the fixed coefficient K, as shown in formula (12).

\[ \forall [v_k, v_l] \in L; K \times \frac{\text{LCM}}{\sum_{i=0}^{n} f_{i \left[v_k,v_l\right]} \_length} = \text{Min\_interval} \]

### 2.2. Algorithm Optimization Based on Load Balancing

This paper introduces soft constraints for the solver to make corresponding solution logic modifications. Soft constraints are based on the requirements of the problem. On the basis that the solution is feasible, additional constraints are imposed on the solution to make it more optimized. These constraints are not required to be met. When the solver determines a conflict, the constraints can be relaxed or deleted.

The additional soft constraint requires that the frame meets the same data stream link, and each frame interval should be greater than a fixed value. The constraint is as shown in formula (10).

### 2.3. Distribution-oriented Iterative Conflict Backtracking Strategy

The algorithm uses step-by-step calculation to divide the message cluster into clusters with fewer messages. When the algorithm generates constraints, each step generates a sub-cluster constraint, which is solved separately. After the result is obtained, the result is placed into the solver according to the constraint and is used as the basis for solving the next sub-cluster. Then moved to the next sub-cluster for solution, until all points are obtained. The cluster is solved.

The step-by-step iteration strategy does not consider the relevance of messages in other clusters. The algorithm solves the cluster in the case where the message position is determined in the previous cluster, and the resulting planning results may be incompatible, which leads to the previous step and this step. Feasible conflict resolution.

The algorithm divides the message cluster into smaller clusters and solves one cluster at a time. When the solution is successful, the result is imported into the solver as a constraint. When the solution fails,
the algorithm traces back to the previous cluster result and solves the previous cluster and the cluster simultaneously; iterative solution, completed after the last cluster.

2.4. Optimized Solver Planning Algorithm Model
The step-by-step constraint is solved as the main part of the message scheduling algorithm. Based on the original constraint-based solver planning algorithm flow, the step-by-step iterative conflict backtrack optimization and the load-balance-oriented enhancement constraint are combined. The specific process is shown in Figure 1 is shown.

The step-by-step algorithm first initializes the context and Solver, which are the interface between the constraint and the solution model provided in the SMT solver. The first step is to construct the solution model; the second step is to segment the message cluster. Firstly, the message is generated to solve the constraint, then the constraint is enhanced according to the strategy. When the solution is successful, the algorithm takes the solution result as a constraint and assists the next solution. Then moved to the next sub-cluster solution, if solving the clustering failure, judge the enhanced constraints and determine whether the constraints can be relaxed. If it can be relaxed, relax the constraint and resolve; if it cannot be relaxed, it means that there is no feasible solution in the current situation, the algorithm needs to roll back the result. Firstly, it is judged whether the cluster is the starting cluster, if it is, it cannot be rolled back. There is no feasible solution for the message. If it is not the cluster, the algorithm rolls back to the solution of the previous step, and expands the cluster to a step length, re-strengthens and continues to solve.

3. Time Triggered Ethernet Planning Algorithm Analysis And Verification

3.1. Algorithm test data set
In the choice of test set, this paper numbers the network nodes and data flow links. The test data also includes a list of messages, and the test set contains only one type of TT message. The message list is shown in Table 1 below. Since the evaluation for the planning algorithm includes the algorithm solution efficiency evaluation and the algorithm load evaluation, in the test set, the control group for the two types of evaluations is also included.
Table 1. Time-triggered Ethernet message list

| Number | Message type | Source node | Destination node | cycle/us | length/kb |
|--------|--------------|-------------|------------------|----------|-----------|
| 1      | TT           | 3           | 0                | 40000    | 100       |
| 2      | TT           | 3           | 2                | 10000    | 400       |
| 3      | TT           | 1           | 2                | 10000    | 400       |
| 4      | TT           | 3           | 4                | 20000    | 300       |
| ...    | ...          | ...         | ...              | ...      | ...       |
| 24     | TT           | 5           | 4                | 10000    | 640       |
| 25     | TT           | 3           | 0                | 40000    | 100       |

3.2. Algorithmic Solution Efficiency Analysis

The algorithm solution time is a direct reflection of the efficiency of the algorithm. In the analysis, the basic solver planning algorithm and the optimization solver planning algorithm are calculated in the case of the same test set of 25, 50, 75 and 100 TT messages. Solve the time, eliminate the difference in results, and get the solution time statistics as shown in the following Table 2:

Table 2. Algorithm solution timetable

| Number of messages | Optimization algorithm solution time | Basic algorithm solution time |
|--------------------|--------------------------------------|-------------------------------|
| 25                 | 8                                    | 6                             |
| 50                 | 218                                  | 315                           |
| 75                 | 851                                  | 1611                          |
| 100                | 2383                                 | 6744                          |

The algorithm solves the timeline diagram as shown in Figure 2:

Figure 2. Two algorithm solution time comparison line chart

It can be seen from the line graph that the optimization algorithm is better than the basic algorithm in solving the problem when the number of messages is large. When the number of messages increases and the solution time increases rapidly, the optimization algorithm grows more gently, which can suppress the complexity of the solution to some extent. And the optimization algorithm is suitable for a large number of messages solving situations in complex networks.
3.3. Algorithmic Load Balancing Analysis

In the load balancing analysis, because the algorithm solution results are compared, in order to reduce the error of the algorithm verification result caused by accidental factors, the message test set of the algorithm uses multiple sets of data to verify the strategy. Under the example of thousands of frames, this paper uses MATLAB to perform scheduling statistics and analysis and verification. The analysis results are as follows:

The network analysis at the network level performs statistics on the messages sent by the entire network, calculates the interval between adjacent messages, and analyses the load balancing indicators. The calculation result is shown in Figure 3.

Figure 3. Network-wide load indicators under three test cases

The load balancing analysis of the whole network counts the standard deviation of the message transmission interval of the whole network under the three test cases. The horizontal axis is the different test group number, and the vertical axis is the value of the load indicator. It can be seen from the figure that in the three test cases, the load algorithm of the optimization algorithm is lower than the basic algorithm, which proves that the algorithm can improve the load balance degree to a certain extent at the whole network level.

According to the analysis of the whole network, nodes and links, the optimization algorithm can achieve certain optimization effects with respect to the basic algorithm, and the load balancing situation at three levels is improved. In addition, the algorithm is based on optimization. The optimization goal makes the optimization algorithm have a more balanced effect among the links. The relative original algorithm has less fluctuation between links and nodes. At the same time, the two algorithms have certain similarities in the trend of each test object, indicating that the two algorithms are different. The core idea is the same. The optimization algorithm mainly changes in the solution strategy. The response to the test data mainly improves the relative basic algorithm on the basis of similar indicators. Finally, in a few cases, the optimization algorithm may be worse than the optimization algorithm. The basic algorithm, on the one hand, shows that the algorithm is not stable enough, and there may be a poor solution. On the other hand, it also shows that the target has the potential to continue optimization, and we should continue to improve.

4. Conclusions

This paper focuses on time-triggered Ethernet message planning, studies the basic constraint-based solver planning algorithm, summarizes the constraints of time-triggered Ethernet message planning according to the SMT solution principle, and proposes load balancing and solution strategy for the shortcomings of the algorithm. Improvements in terms. At the same time, the optimization effects of the algorithm in terms of solution efficiency and network load are analysed, and the effectiveness of the algorithm is determined. According to the results of analysis and verification, the constraint-based solver optimization algorithm can basically meet the requirements of Ethernet programming and solving. At
the same time, compared with the basic algorithm, the algorithm has improved to some extent in terms of efficiency and load balance. A certain effect has been achieved. In the next research, we will continue to improve the network constraints, and at the same time concrete the parameters in the constraints to improve the maturity of the algorithm.

References

[1] Real J, Sáez S, Crespo A. Combining Time-Triggered Plans with Priority Schedul-ed Task Sets[J]. 2016.
[2] Craciunas S S, Oliver R S, Chmelik M, et al. Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks[C]// International Conference on Real-time Networks & Systems. 2016.
[3] Steiner W, Steiner W. Design optimization of TTEthernet-based distributed real-time systems[M]. Kluwer Academic Publishers, 2015.
[4] Dvorak J, Heller M, Hanzalek Z. Makespan minimization of Time-Triggered traffic on a TTEthernet network[C]// 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS). IEEE, 2017
[5] Trindade A B, Cordeiro L C. Applying SMT-based verification to hardware/software partitioning in embedded systems[J]. Design Automation for Embedded Systems, 2016, 20(1):1-19.
[6] Xu Xiaofei, Cao Chen, Guo Jun, et al. TT-RMS: Communication table generation algorithm of time-triggered network[J]. Journal of Beijing University of Aeronautics and Astronautics, 2015, 41(8):1403-1408.
[7] Zhang Chao, Nan Jianguo, Chu Wenkui, et al. An Improved Communication Table GenerationAlgorithm Based on Time-Triggered Rate Monotonic Scheduling[J]. Journal of Air Force Engineering University(Natural Science Edition) , 2016, 17(6):82-87.