Design and Performance Testing of a Simulation Model for Time-Triggered Ethernet

Bidong Duan
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: 1145597333@qq.com

Jing Cheng
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: chengjing@xatu.edu.cn

***Abstract*—Time-Triggered Ethernet (TTEthernet) is a new hybrid transmission network technology that introduces time-triggered, synchronization, and security mechanisms based on traditional Ethernet. It is difficult to verify the overall performance mechanism of the TTEthernet through manual calculation. Therefore, this paper models the TTEthernet End System and Switch, and simulates the overall performance of the TTEthernet Model based on the OPNET platform. Firstly, this paper introduces the key mechanisms and development status of the TTEthernet. Secondly, the simulation modeling of the TTEthernet End System and Switch are described, and the key technologies within the model are explained; The End system is an important component of the network, responsible for data generation, transmission, and processing. The Switch is responsible for data forwarding and routing, which plays a crucial role in the performance and stability of the network. Finally, the performance of the TTE network model in this article is analyzed through simulation data, verifying the timeliness, determinacy, and reliability of the model. Simulation data shows that the global throughput of the TTE network model in this article is stable at around 70Mbps, the End-to-End Delay of TT services is around 2.5ms, and the packet loss rate is almost zero, which meets the expected results. However, RC and BE services have higher end-to-end latency and packet loss rates than TT services due to their event triggered messages, but they still meet expectations.**

Keywords-Time-Triggered Ethernet; Simulation Model; Switching Architecture; OPNET

I. INTRODUCTION

A. Research Motivation and Significance

Time-Triggered Ethernet (TTEthernet) is a real-time communication protocol based on Ethernet technology. It uses time-triggered transmission to send data at predetermined intervals, ensuring the real-time and reliable delivery of data. TTEthernet can support various applications, including aerospace, military, and industrial automation. Its main features include high reliability, low latency, high bandwidth, and strong scalability.

In order to better carry out the business planning and maximum carrying capacity evaluation of various network nodes, the 503 Institute of the Fifth Academy of Aerospace Science and Technology has carried out a horizontal project called "TTEthernet Simulation and Calculation Development Tool" in recent years. The TTEthernet simulation and calculation development tool aims to provide support for product design, and is a network planning tool based on the OPNET simulation platform. The tool adopts advanced simulation algorithms and models, which can simulate the data transmission and processing process in various network environments, so as to accurately evaluate the performance and reliability of network nodes.

The First Aircraft Design and Research Institute of China Aviation Industry Corporation (AVIC 1) has developed a Time-Triggered FC Network Simulation Model based on the OPNET simulation platform in order to further improve the determinacy and reliability of aviation and aerospace networks, and meet the high-security requirements of networks. The Time-Triggered FC Network is based on the FC network and incorporates technologies such as TTEthernet and AFDX for traffic control, margin management, and time triggering, which further improves the

DOI: 10.2478/ijanmc-2023-0066

determinacy and reliability of the network. The time-triggered FC network simulation platform can assist system designers in the initial top-level design stage of the system to quickly simulate the entire network topology and network traffic, and verify and iterate through digital simulation to validate the ICD design.

B. Related Work

This article proposes a simulation model design for TTEthernet, elaborating in detail the core functional design of the End System Model and the Switch Model, and conducting simulation tests on the model based on the OPNET platform.

When designing the simulation of the TTEthernet Model, this article first carried out the overall model design to ensure the completeness and feasibility of the model. Then, the design ideas of the End System Model were explained in detail, including data stimulation, scheduling and transmission mechanism, and redundancy strategy. Finally, this article designed a 24-port Switch Model, and provided a detailed explanation of the various switching architectures designed in the Switch, making the Switch Model more suitable for practical application scenarios.

After completing the simulation model design, this article implemented the model simulation based on the OPNET platform and conducted simulation tests on the throughput, End-to-End Delay, and packet loss rate performance of the TTEthernet Model proposed in this article based on certain initial data. Through the summary of experimental test data, this article draws a conclusion that proves the feasibility and effectiveness of the TTEthernet simulation model design proposed in this article.

II. Research Status at Home and Abroad

Time-Triggered Ethernet (TTEthernet) is a network communication technology based on time synchronization, which enables high-precision data synchronization and communication. Currently, TTEthernet has been widely researched and applied both domestically and internationally. TTEthernets have outstanding advantages in real-time performance, determinism, reliability, and bandwidth protection. [1] TTEthernet has been applied in NASA's manned space shuttle project and has attracted the attention of foreign aircraft manufacturers. China is also considering adopting TTEthernet technology in future spacecraft systems. [2]

In foreign countries, Time-Triggered Ethernet has been widely used in industrial automation, robot control, and medical equipment, aerospace and other fields. For example, the European Space Agency uses Time-Triggered Ethernet to achieve high-precision synchronization of satellite communication and navigation systems; Siemens in Germany applies Time-Triggered Ethernet to industrial automation and intelligent manufacturing; and the aerospace and military equipment industries in Europe and the United States have begun to deploy TTEthernet with good results. [4]

In China, research and application of Time-Triggered Ethernet are constantly developing. For example, the Institute of Automation of the Chinese Academy of Sciences has developed a high-precision data acquisition system based on Time-Triggered Ethernet, which can achieve microsecond-level data synchronization and acquisition. Shanghai Jiao Tong University has also conducted research on the application of Time-Triggered Ethernet in robot control and intelligent manufacturing. [7]

In general, Time-Triggered Ethernet has been widely researched and applied both domestically and internationally. In the future, with the development of industrial automation and intelligent manufacturing, the application prospects of Time-Triggered Ethernet will become even broader.

III. Design of Simulation Models

A. Overall Design

The TTEthernet overall model consists of the End System Model, Switch Model, and Link Model. The TTEthernet overall model design is shown in Figure 1.

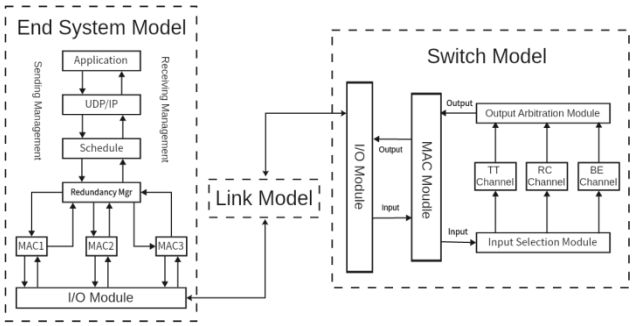

Figure 1. TTEthernet Overall Model

The End System Model is responsible for sending and receiving processing of the three types of data (TT, RC, BE) in the TTEthernet. The Switch Model is responsible for forwarding the three types of data. The Link Model is a point-to-point full-duplex link, responsible for implementing bidirectional transmission of data frames between the End System and the Switch.

B. Design of End System Model

The End System Model of Time-Triggered Ethernet (TTEthernet) consists of six modules: Application Module, UDP/IP Module, Scheduling Module, Redundancy Management Module, End System MAC Module, and I/O Module. These modules work together to implement the various protocol contents of the TTEthernet. The TTEthernet End System Model based on OPNET is shown in Figure 2, where the arrows between modules represent the flow of three types of data and the interaction of information.

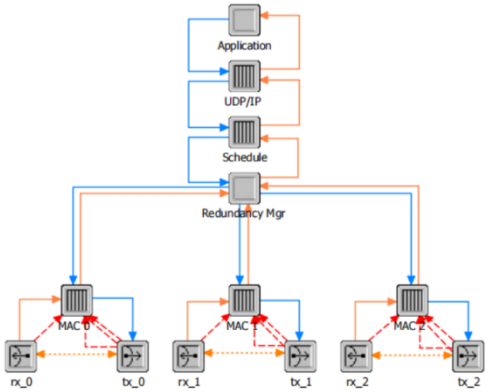

Figure 2. TTEthernet End System Simulation Model

The frequency and conditions of message transmission are the first issues to be addressed in the TTEthernet End System simulation model. To achieve this, the association between tasks and messages must first be established for each application. The business of the application is composed of a series of messages, and the completion of tasks relies on the sending and receiving of messages. Therefore, the correspondence between tasks and messages can be described by the following equation.

$$T_k = \{t_1, t_2, \cdots, t_m\} \tag{1}$$

The formula represents a certain task on the application of the End System, denoted as T_k, which consists of a set of m messages. Each message is defined by a five-tuple:

$$t_i = \{I_{ti}, V_{ti}, L_{ti}, P_{ti}, T_{ti}\} \tag{2}$$

In the formula:

I_{ti} Represents the message sequence number, which is unique;

V_{ti} Represents the virtual link number of the message, which represents the virtual link that the message passes through in the TTEthernet;

L_{ti} Represents the frame length of the message and the message frame length are in bytes;

P_{ti} Represents the generation frequency or period of the message, using various function excitation methods to simulate non-periodic message events to be closer to the real network communication model;

T_{ti} Represents the message type, which is used to define the TT, RC, and BE message types.

After the TTEthernet message is encapsulated into an IPv4 data frame by the End System, it needs to enter the corresponding cache queue and wait for scheduling. The maximum carrying capacity of the cache queue is determined by the queue's packet capacity and bit capacity. Time-Triggered messages (TT) are scheduled based on the concept of time windows, which specify the start time (Tx_{start}) and end time (Tx_{end}) for the message to be sent within an integration cycle

(Tx_{period}). Therefore, if it is allowed to schedule the message at the current time (T_N), the following conditions must be met:

$$\begin{cases} Tx_{start} \le (T_N) \bmod (Tx_{period}) < Tx_{end} \\ L_{ti} / Pt_{rate} < Tx_{end} - (T_N) \bmod (Tx_{period}) \end{cases} \quad (3)$$

"Pt_{rate}" Refers to bandwidth.

For the scheduling of event-triggered messages (RC and BE), it is required to be completed within the time interval between two time windows. In order to achieve bandwidth management while scheduling TT messages based on time window order, two attribute values, L_{max} and BAG, need to be set for each virtual link. In ordinary Ethernet, due to the lack of L_{max} and BAG restrictions, the network traffic sent by End Systems is relatively random, which can easily cause surge traffic, leading to congestion or even congestion of Switches, increasing the queuing time and delay of data frames, and having a high degree of uncertainty. In TTEthernets, the BAG value is applied to the scheduling of RC messages.

BAG (Bandwidth Allocation Gap): defines the minimum time interval for continuously sending RC messages on the same link. After sending a message at the sending end, if another RC message is received within the BAG time, the next message can only be sent after waiting for the BAG time.

Due to the performance differences among different scheduling strategies, they may have different impacts on the delay of data frames. In the actual scheduling process, the scheduler should try to keep the BAG and L_{max} values set for each virtual link unchanged. However, due to the introduction of the scheduler and scheduling strategies, there will be a delay time for data frames waiting for scheduling, which introduces the concept of jitter. In Time-Triggered Ethernet, jitter refers to the time interval between starting to send a data frame within the BAG value range and actually sending it. Different scheduling mechanisms may produce different time intervals. Figure 3 shows three different jitter scenarios.

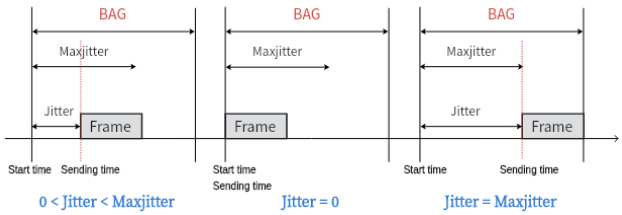

Figure 3. Three types of jitter situations

The jitter of each End System should satisfy the following formula:

$$Jitter \le 40 + \frac{\sum_{Set\ of\ VLs} (20 + L) \times 8}{Pt_{rate}} \quad (4)$$

Due to the high reliability requirement for data transmission in TTEthernet, this model adopts a three-redundancy fault-tolerant management mechanism. The entire TTEthernet consists of three identical networks A, B, and C. Each End System contains three Ethernet interfaces A, B, and C, which are connected to the Switch ports in networks A, B, and C respectively, forming a three-redundancy network composed of networks A, B, and C. When an End System sends a data frame on a single virtual link, it duplicates the data frame into three copies and sends them to networks A, B, and C simultaneously. The data frames are transmitted independently in networks A, B, and C without affecting each other. The receiving end will receive the same data frame from networks A, B, and C on the virtual link. The three-redundancy network structure is shown in Figure 4.

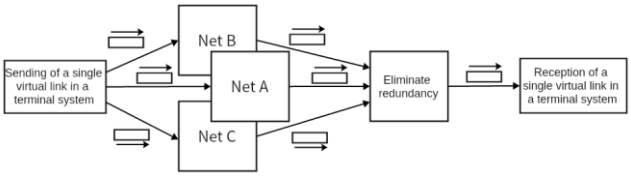

Figure 4. Triple Redundancy Network Architecture

The redundancy processing module checks the data frames submitted by networks A, B, and C, and adopts the principle of first come, first served. By comparing the frame numbers, redundant data frames are removed to maintain the uniqueness of frame numbers, thus achieving redundancy management. In the event of a lost data frame in one network, data frames from another network

can be used to correct it, greatly enhancing the reliability of data transmission.

C. Design of Switch Model

The Switch Model of Time-Triggered Ethernet can be divided into five modules: I/O Module, Switch MAC Module, Input Selection Module, Message Channel Module, and Output Arbitration Module. The main functions of the Switch are as follows:

1) Forward data to the corresponding output port based on the configuration information of the VL to ensure the order of data transmission for the same VL;

2) Support unicast, multicast, and broadcast based on the configuration information of the VL;

3) Support the reception, scheduling, and transmission of three types of business messages: TT (time-triggered), RC (rate-controlled), and BC (best-effort).

The TTEthernet Switch adopts different switching architectures for the three types of messages. The TTEthernet Switch simulation model based on OPNET is shown in Figure 5.

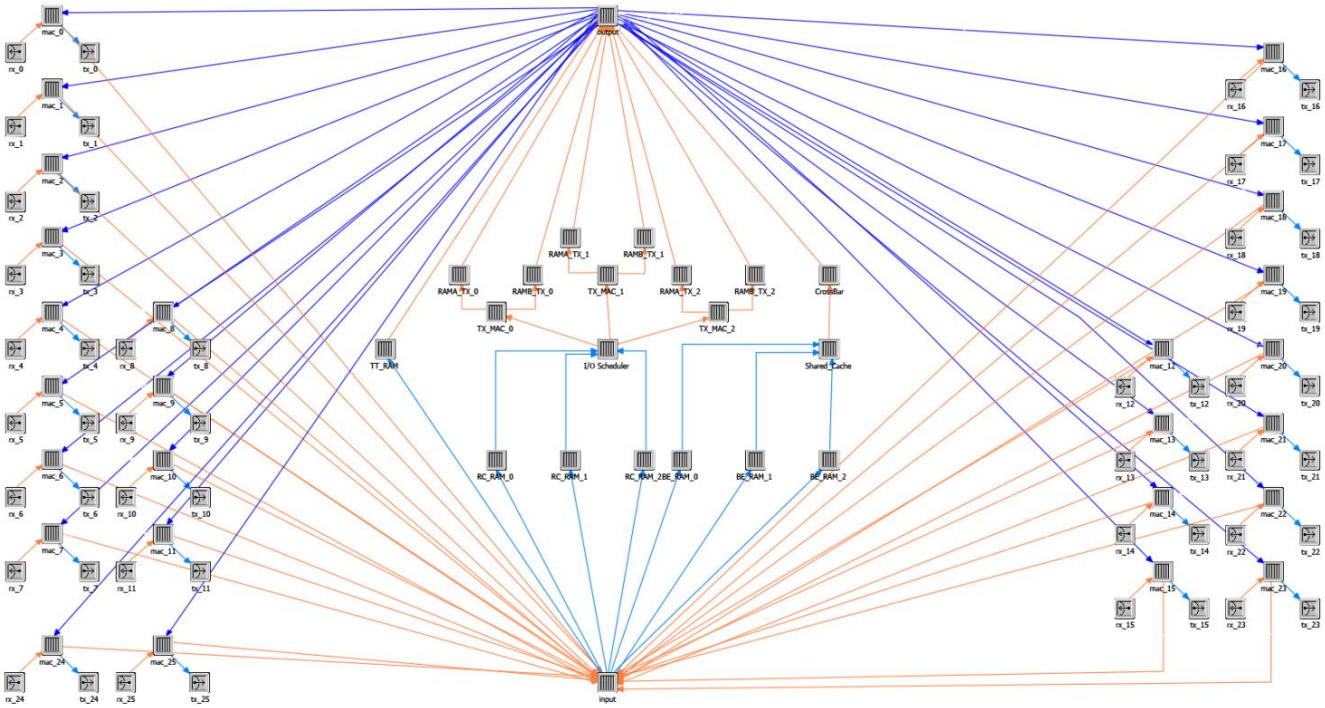

Figure 5. TTEthernet Switch Simulation Mode

For all types of messages that arrive at the Switch, a FULL-MESH architecture is used as the exchange channel for TT messages in order to ensure real-time, deterministic, and reliable TT message delivery to the greatest extent possible. FULL-MESH is a network connection architecture, also known as a fully connected architecture, in which all nodes are directly connected, ensuring that a direct path can be found between any input port and any output port.

The RC channel is an exchange architecture consisting of a set of shared buffers and three ping-pong buffers. Each set of ping-pong buffers consists of an input data controller and two identical RAMs. The operation structure of the ping-pong buffers is shown in Figure 6.

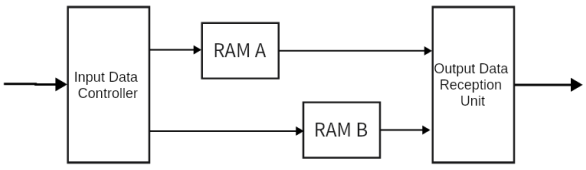

Figure 6. Ping-Pong buffer operation structure

The input data stream is divided equally into two data buffers by the input data selector controller. In the first buffer cycle, the input data frame is stored in RAM A. In the second buffer cycle, the next data frame is stored in RAM B by switching the input data controller, while the data frame cached in the first buffer cycle of RAM A is transferred to the output data receiver unit. In the third buffer cycle, the input data frame is stored in RAM A while the data frame cached in RAM B is transferred to the output data receiver unit by switching the input data controller. This cycle continues, achieving the processing of high-speed data streams using a low-speed data preprocessing module.

The cache architecture of the BE channel adopts a shared cache combined with a CrossBar structure. CrossBar is also known as a crosspoint switch or a matrix switch, which can effectively compensate for some of the shortcomings of shared memory mode. The CrossBar switch matrix is shown in Figure 7.

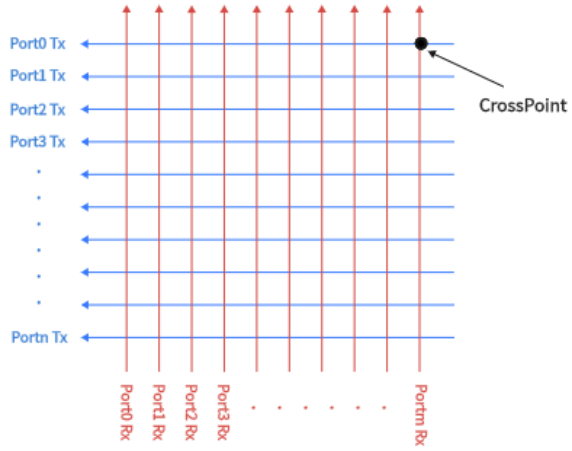

Figure 7. CrossBar Switching Matrix

The Crossbar architecture is a three-level architecture that consists of a switch matrix. Each CrossPoint is a switch, and the switch controls the forwarding of input to a specific output. The Crossbar is non-blocking internally (relatively speaking). As long as multiple crosspoints are closed simultaneously, multiple different ports can transmit data simultaneously. In this sense, we consider all Crossbars to be non-blocking internally because they can support all ports to exchange data at line speed simultaneously.

Each CrossPoint has a cache queue for buffering data frames, with a capacity equal to the size of the two longest data frames in the network. During data output, a round-robin algorithm is used to sequentially poll each CrossPoint under the same output port. When a data frame is present in the node's cache, the frame information is read and the data frame is output. The output scheduling process involves checking whether the output port is idle and whether the CrossPoint cache is readable when a data frame enters the CrossPoint cache. If the conditions are met, the round-robin scheduling algorithm is used to poll the destination port number that needs to be dequeued, and the data frame is output accordingly. After data output is complete, the output port is set to idle.

IV. Simulation Test Results and Analysis

The behavior between the End Systems, switches, and links in the TTEthernet simulation system implemented based on OPNET can be represented by the timing diagram shown in Figure 8.

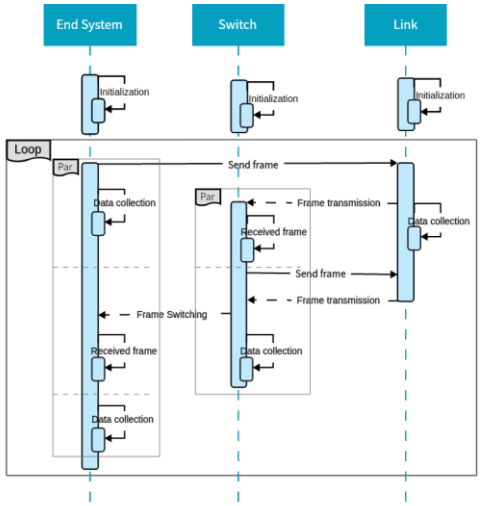

Figure 8. TTEthernet Simulation System Timing Diagram

After having sufficient initialization data support, the entire simulation network will continuously perform loop processes such as sending, forwarding, and creating and destroying data frames, data recording, error handling, etc. until the simulation is completed. After the simulation is executed, the network's statistical

module will display the simulation test results, making it easier to evaluate the performance of the entire network.

This article conducts simulation tests on the network topology shown in Figure 9.

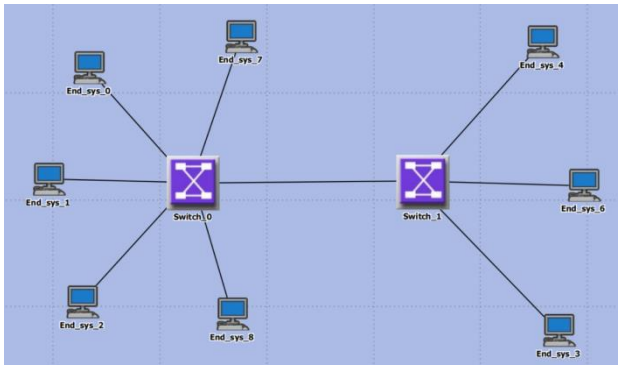

Figure 9. TTEthernet topology

The TTEthernet simulation test results require input of configuration information. Figure 10 shows some of the configuration information during the simulation test process.

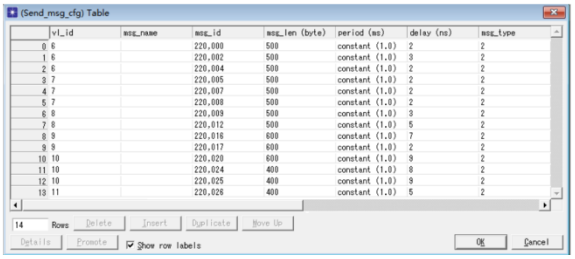

Figure 10. Simulation configuration information

The initial amount of data obtained from each simulation is often very large, so the concept of time bucket is introduced to process the simulation data properly before presenting the final results. In OPNET's simulation data processing, one time bucket is often defined as one percent of the total simulation time, denoted as T_B in this article, and the following test results can be described accordingly.

A. Throughput

Assuming that the TTEthernet simulation system received n data frames within a certain time period, the throughput can be calculated using the following Equations.

$$R_t^F = \frac{(L_T + L_1)(T_1 - \lfloor T_1 / T_B \rfloor \times T_B)}{T_1 \times T_B} \quad (5)$$

$$R_t^M = \sum_{j=2}^{n-1} \frac{\left(L_T + \sum_{i=1}^{j} L_i\right)(T_j - T_{j-1})}{T_j \times T_B} \quad (6)$$

$$R_t^L = \frac{\left(L_T + \sum_{k=1}^{n} L_k\right)(\lceil T_n / T_B \rceil \times T_B - T_n)}{T_n \times T_B} \quad (7)$$

$$R_t^n = R_t^F + R_t^M + R_t^L \quad (8)$$

R_t^n Represents the throughput value that the simulation system needs to record at time t; L_t represents the total amount of data received by the current simulation system; T_n represents the simulation time when the system receives the message; L_i represents the length of the data frame received at time T_i. The throughput of the TTEthernet simulation system obtained by this calculation method is shown in Figure 11, and from the test results, the throughput of the entire network is stable at around 70Mbps,indicating that the network model has good data transmission ability and can meet the needs of most application scenarios.

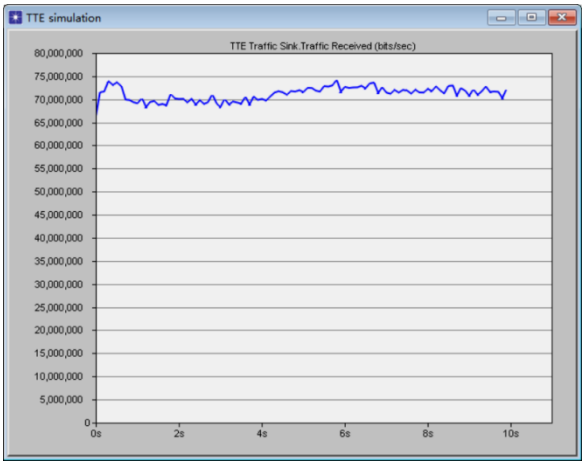

Figure 11. Throughput of TTEthernet simulation system

B. End-to-End Delay

End-to-End Delay refers to the total time it takes to send data from one network endpoint to another and receive confirmation. The key factors that affect end-to-end delay in the TTEthernet simulation system are as follows:

1) Frame encapsulation delay;

2) Cache size and the speed of data frames entering and leaving the cache;

3) Queuing delay, which refers to the time it takes for data to be stored in the buffer queue of network devices such as routers or switches waiting for processing, and is related to network congestion and buffer size;

4) Delay in scheduling policies executed by End Systems and switches;

5) Port transmission rate, the time it takes for the transmitter to push the message onto the link;

6) Link propagation rate, the transmission rate of messages on the cable;

7) Delay in performing data frame verification;

8) Interval for Switch polling of data frames;

9) Delay in executing redundancy policies;

10) Number of switches in the network, too many switches will increase the number of hops for data frames, and too few switches will cause internal congestion in the switches;

11) Delay in the internal cache architecture.

The TTEthernet simulation system will automatically simulate the above-mentioned delay. Therefore, after receiving a data frame, the formula for calculating the End-to-End Delay of the data frame is as follows:

$$End\text{-}to\text{-}End\ Delay = Frame.T_N - Frame.T_C \quad (9)$$

In the formula, $Frame.T_N$ represents the time when the receiving end receives the data frame, and $Frame.T_C$ represents the time when the data frame is created. Therefore, after the simulation is completed, the End-to-End Delay of the three types of messages in the TTEthernet simulation system can be obtained as shown in Figure 12.

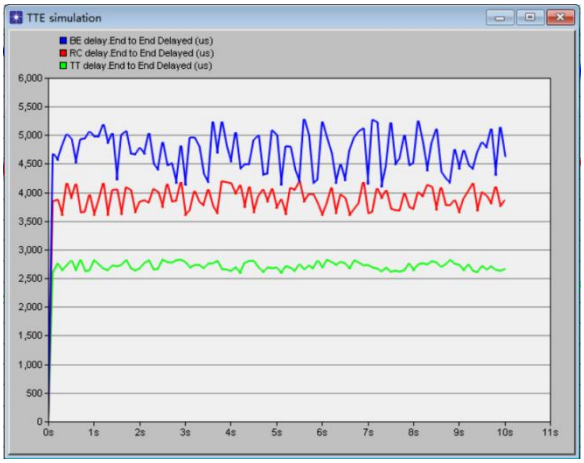

Figure 12. End-to-End Delay of three types of messages

Due to the fact that time-triggered (TT) messages are scheduled and sent based on time windows and have the highest scheduling priority, the End-to-End delay of TT messages is the smallest among the three types of messages and the fluctuation of the delay curve is also relatively small. Best-Effort (BE) messages, as messages sent with best effort, have the lowest real-time performance and reliability, resulting in the largest fluctuation of the End-to-End Delay curve, which is slightly higher than that of RC messages in terms of numerical value.

C. Packet Loss Rate

The packet loss rate refers to the ratio of the number of lost data packets to the number of packets sent during transmission. The packet loss rate is related to configuration information such as packet length, packet transmission frequency, module queue size, and time window planning. The steps to calculate the packet loss rate in OPNET are:

1) Define a variable PkCreate within the End System Model to record the total number of data frames generated by the data source;

2) Define a variable PkSend to record the total number of data frames finally sent out from the model;

3) The equation for calculating packet loss rate (P_L) is (10).

$$P_L = \frac{PkCreate - PkSend}{PkCreate} \quad (10)$$

However, during the simulation process, multiple data packets may be lost at the same time, and OPNET needs to use a time bucket mode for weighted averaging internally. Assuming that m packets are lost by a terminal system within one bucket time, the final equation for calculating the packet loss rate is (11).

$$P_{LossRate}^{m} = \frac{\sum_{i=1}^{m} P_L^i (T_i - T_{i-1})}{T_B} \tag{11}$$

In the equation, $T_0 = \lfloor T_1 / T_B \rfloor \times T_B$.

Based on the configuration information input into the TTEthernet simulation system, the packet loss rate of the TTEthernet simulation system can be obtained as shown in Figure 13.

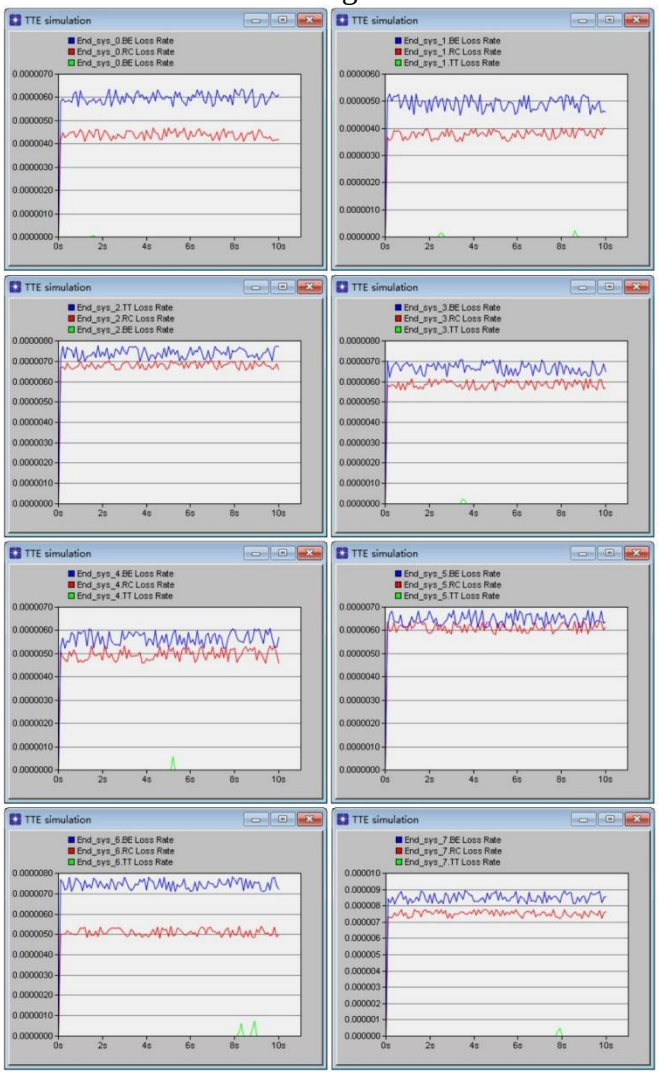

Figure 13. Packet loss rates for three types of messages

According to the simulation test results, the packet loss rate of time-triggered messages (TT messages) in the entire TTE network is almost 0, which means that the transmission of time-triggered messages is very stable and reliable. This highly reliable transmission is critical for time-triggered applications that require precise synchronization and determinism. In addition, the packet loss rate of RC messages and BE messages is also controlled below 1 in 10,000, which indicates that these two types of messages also have quite high transmission reliability and stability.

These simulation results further verify the reliability and stability of the TTE network model, and provide strong support for its practical application.

V. CONCLUSIONS

This article focuses on simulating and modeling Time-Triggered Ethernet and conducting simulation testing and analysis of network performance based on OPNET. In the End System Model, the timeliness, determinacy, and reliability of time-triggered messages are ensured through time window-based scheduling and the strategy of three redundant networks. In the Switch Model, different exchange architectures are used as message channels for different types of messages to ensure the orderliness of various messages and reduce internal congestion in the Switch. In addition, this article also simulated and modeled the TTEthernet Model based on OPNET and analyzed the overall performance of the network through experimental data.

Network simulation plays an important role in evaluating network design and optimizing network performance. Through network simulation, different network designs and solutions can be evaluated and tested before actual deployment, which can greatly save time and cost in design and deployment. It can also simulate various complex network topologies and environments, including different types of devices, protocols, and traffic volume. This allows network designers to fully understand the various possibilities and fundamental factors of network operation.

This article provides some reference suggestions for the future design of Time-Triggered Ethernet, but there are still some aspects that can be improved in the methods proposed in this article. For example, on switches with more ports, the CrossBar switching architecture may no longer be the optimal choice, and it may be considered to use CLOS architecture or other architectures to optimize the message channels of switches in the future.

REFERENCES

[1] L. Zhang, D. Goswami, R. Schneider, and S. Chakraborty, "Task- and network-level schedule co-synthesis of Ethernet-based time-triggered systems," in 2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC), Singapore, Jan. 2014. doi: 10.1109/aspdac.2014.6742876.

[2] P. Tabuada, "Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks," in IEEE Transactions on Automatic Control, vol. 52, no. 9, pp. 1680-1685, Sept. 2007, doi: 10.1109/TAC.2007.904277.

[3] J. Li, H. Xiong, Q. Li, F. Xiong, and J. Feng, "Run-Time Reconfiguration Strategy and Implementation of Time-Triggered Networks," Electronics, vol. 11, no. 9, p. 1477, May 2022, doi: 10.3390/electronics11091477.

[4] P. Muoka, D. Onwuchekwa, and R. Obermaisser, "Adaptive Scheduling for Time-Triggered Network-on-Chip-Based Multi-Core Architecture Using Genetic Algorithm," Electronics, vol. 11, no. 1, p. 49, Dec. 2021, doi: 10.3390/electronics11010049.

[5] L. Behera, "A fault-tolerant time-triggered scheduling algorithm of mixed-criticality systems," Computing, vol. 104, no. 3, pp. 577–599, Mar. 2022, doi: 10.1007/s00607-021-01026-5.

[6] J. H. Goh and M. K. Ishak, "Implementing Time-triggered Communication for Embedded Ethernet Real-time Network using IEEE 1588 Time Synchronization," IEIE Transactions on Smart Processing & Computing, vol. 10, no. 5, pp. 431–438, Oct. 2021, doi: 10.5573/ieiespc.2021.10.5.431.

[7] H. Yuan, T. Zheng, K. Zhang, and Y. Wang, "An Efficient Schedule Synthesis Method based on Constraint Programming Technology for Time-Triggered Ethernet," in 2020 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, May 2020. doi: 10.1109/icccs49078.2020.9118453.

[8] Z. Zheng, F. He, and H. Xiong, "Routing Optimization of Time-Triggered Ethernet Based on Genetic Algorithm," in 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, Oct. 2020. doi: 10.1109/dasc50938.2020.9256594.

[9] B. Pavkovic, M. Sandic, and N. Teslic, "A genetic simulation strategy: Application to single-fault analysis of TTEthernet synchronization protocol," Journal of Systems Architecture, vol. 117, p. 102169, Aug. 2021, doi: 10.1016/j.sysarc.2021.102169.

[10] X. Ma and A. Hamdulla, "Hybrid Scheduling Technology of Time-triggered Ethernet Switches: A Review," Journal of Physics: Conference Series, vol. 1673, no. 1, p. 012024, Nov. 2020, doi: 10.1088/1742-6596/1673/1/012024.

[11] J. Lu, H. Xiong, F. He, Z. Zheng, and H. Li, "A Mixed-Critical Consistent Update Algorithm in Software Defined Time-Triggered Ethernet Using Time Window," IEEE Access, vol. 8, pp. 65554–65565, Jan. 2020, doi: 10.1109/access.2020.2984749.

[12] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Routing and Scheduling of Time-Triggered Traffic in Time-Sensitive Networks," IEEE Transactions on Industrial Informatics, pp. 4525–4534, Jul. 2020, doi: 10.1109/tii.2019.2950887.

[13] X. Yang, Y. Huang, J. Shi, and Z. Cao, "A Performance Analysis Framework of Time-Triggered Ethernet Using Real-Time Calculus," Electronics, vol. 9, no. 7, p. 1090, Jul. 2020, doi: 10.3390/electronics9071090.

[14] L. Zhao, M. Ni, X. Cui, W. Wang, and J. Zhang, "The Research of Scheduling Algorithm Based on Time-Triggered Ethernet," in 2020 22nd International Conference on Advanced Communication Technology (ICACT), Phoenix Park, PyeongChang,, Korea (South), Feb. 2020. doi: 10.23919/icact48636.2020.9061364.