An approach for integrity assurance in Time-triggered Ethernet

Pengyu Chen¹, Zhaobin Li² and Weichen Lian*¹

¹ Department of Electronics and Communication Engineering, Beijing Electronic Science & Technology Institute, Fengtai, Beijing, 100070, China
² Department of Electronics and Communication Engineering, Beijing Electronic Science & Technology Institute, Fengtai, Beijing, 100070, China
*Corresponding author’s e-mail: kevinbaylor@163.com

Abstract. In communication systems, integrity is the ability of a system to detect faults and achieve a fail-safe state during operation, and is one of the important attributes of credibility. Currently, there is no effective integrity assurance mechanism in Time-Triggered Ethernet (TTE). This paper combines the principles of cryptography, designs TTE switch with integrity guarantee mechanism, and builds a simulation platform to verify the design implementation. The simulation results show that the mechanism effectively implements the data integrity guarantee of TTE and improves the reliability of the TTE system.

1. Introduction

Ethernet has continued to expand its coverage and enter new applications since its inception. However, in the initial development and application of Ethernet, the real-time requirements of the network were not considered. Time-triggered Ethernet has attracted more and more attention due to its high bandwidth, low cost, and compatibility with traditional Ethernet devices.

TTE is a time-triggered protocol-based communication network developed on the basis of standard Ethernet. It integrates standard Ethernet protocol, AFDX protocol and time-trigger protocol, and has the advantages of standard Ethernet and time-triggered communication network.

At present, the time-triggered Ethernet lacks an efficient and reliable data integrity guarantee mechanism. The data integrity guarantee mechanism in this paper aims to provide a safe and reliable data transmission environment for time-triggered Ethernet. This paper realizes the data integrity guarantee mechanism of time-triggered Ethernet by studying real-time network transmission mechanism, data integrity verification algorithm and design time-triggered Ethernet switch.

2. Related work

Foreign research on time-triggered Ethernet has started earlier than domestic ones and has been applied to industrial practical fields. In 2005, Professor Kopetz proposed the design idea of time-triggered Ethernet [1]. In 2014, the Institute of Real-Time Computer Systems at the Technical University of Munich, Germany, unified the task scheduling and network scheduling through the Integer Programming method to perform application level scheduling [2]. In recent years, there have been some foreign institutions engaged in time-triggered Ethernet research. AS6802 TTEthernet [3], developed by the Society of Automotive Engineers (SAE), is widely supported and has been adopted by NASA. Recently, most of the research on TTE is mainly on time synchronization [4], time scheduling [5] and...
switch design [6]. The above research work is mainly based on theoretical analysis and experimental simulation design of colleges and universities.

At present, there is a lack of efficient and reliable data integrity guarantee mechanism in Time-triggered Ethernet. Jia Mengyuan uses switches to build dual redundant network backbone links to provide integrity guarantee for data packet physical transmission. Compared with Jia’s, this paper aims to design a time-triggered Ethernet data integrity assurance system based on cryptography. The focus of this paper is on the research and design of time-triggered Ethernet key algorithms and components. Compared with related research at home and abroad, this solution has the advantages of light weight and reliability.

3. System description

3.1. TTE switch design

As shown in Figure 1, is the TTE switch function design, which consist of several modules. The central controller module is mainly responsible for time synchronization and integrated management. The receiving control module is primarily for real-time packets, ensuring that only real-time packets arriving between receive windows are received. The sending control module is mainly responsible for querying the static routing table, and sending the corresponding data packet through the corresponding output port. The digest generate module and the digest authentication module make sure that the integrity is in assurance.

3.2. System framework

As shown in Figure 2, data transmission.
Figure 2. depicts the progress of data processing of the TTE switch. The specific process is as follows.

1. The data enters the switch to determine whether it is a TT message. If it is not a TT message, the direct send process is performed; if it is a TT message, step 2 is performed;

2. Determine the location of the switch. If it is an intermediate switch, perform the direct send process. If it is an ingress switch, go to step 3. If it is an egress switch, go to step 4.

3. The ingress switch performs the digest generation, summarizes the data entering the switch, and adds the digest value as a data packet to the sending period;

4. The egress switch performs digest verification, summarizes the original data packet, and compares the consistency of the digested values before and after;

### 3.2.1. Receiving control module.

When the receiving control module receives the data packet, determine whether it is a TT message. If it is a TT message and arrives in the specified receiving window, send the message. Otherwise, drop message so as not to cause delays and other effects on other real-time tasks.

### 3.2.2. Digest generate module.

Receive a message sent from the receiving control module and determine whether the switch is an ingress switch. If it is an ingress switch, and the message is a TT message, generate a digest value, and send the message to the central controller module. Otherwise, send the message directly.

The main function of the module is to calculate the digest value for the TT message entering the ingress switch. Follow the process below:

1. Determine whether it is a TT message, and each packet is further divided into 16 32-bit sub-packets. The output of the algorithm consists of four 32-bit packets that are concatenated to form a 128-bit hash value. The specific process is:

   The message sequence \([T'_1 , T'_2 , \ldots , T'_M]\) is filled so that the bit length is 488 when mod 512, and the total length of the data \(L_1\) is

   \[ L_1 = N \times 512 + 488 \quad (N \text{ is a positive integer}) \]  

   The method of filling is to fill a 1 and several 0 after the information until the above conditions are met.

   Then add 64 bits of the original message length after the result, so that the message length is exactly an integer multiple of 512.

   \[ L_2 = N \times 512 + 488 + 64 = (N + 1) \times 512 \quad (N \text{ is a positive integer}) \]

2. Then generate the digest

   \[ T'_{M+1} = \text{Hash}(L_2) \]

### 3.2.3. Central controller module.

The main function of Central controller module is to send different types of data packets. For the received data packet, if it is a real-time data packet, send directly; for non-real-time data packets, in the process of sending, it is necessary to ensure that even if the non-real-time data packet is preempted, it can be sent again.

### 3.2.4. Digest Authentication module.

Receive message and determine whether the switch is an egress switch. If it is an egress switch and the message is a TT message, calculate a digest value, and verify the consistency of the digest value. If the verification is consistent, send the message to the send control module, otherwise, discard the message. If it is not an egress switch, send the message directly.

The main function of Digest Authentication module is to verify the consistency of the digest value. Follow the process below:
The M+1 TT message transmission cycle entering the digest verification module is arranged in an ascending order to obtain a message sequence.

\[ [T'_1, T'_2, \ldots, T'_{M+1}], \quad T'_{i+1} \geq T'_i, \quad (i \in [1, M-1]) \]  

(6)

Take the top M messages get the message sequence

\[ [T'_1, T'_2, \ldots, T'_M], \quad T'_{i+1} \geq T'_i, \quad (i \in [1, M-1]) \]  

(7)

Generate the digest, and the digest algorithm H is identical to the digest algorithm in the digest generating module, and a new digest message \( T''_{M+1} \) is obtained.

\[ T''_{M+1} = H([T'_1, T'_2, \ldots, T'_M]) \]  

(8)

Comparing the consistency of the message \( T'_{M+1} \) with \( T''_{M+1} \), if the consistency is successful, the integrity check is successful, and the data is sent to the target node; if not, the verification fails and the message is dropped.

3.2.5. *Sending control module.* The main function of sending control module is to distribute the data packets. Different output ports are connected to different network nodes, and various types of data packets that are scheduled are forwarded in the transmission control processor to the corresponding next network node.

4. Simulation and performance

4.1. *Simulation environment*

- C++ sources
- MSG files
- opp_msgc
- *_m.cc/h files
- Simulation kernel and user interface libraries
- Compiling and linking
- Simulation program
- Ned files
- Running
- Result files

Figure 3. Simulation principle of OMNeT++.

![Simulation principle of OMNeT++](image)

Figure 4. Simulation topology.
OMNeT++ is a C++-based, object-oriented discrete-time simulation tool [8] for simulation of communication networks, multiprocessor systems, and other distributed systems [9]. As shown in Figure 3, by compiling and linking C++ sources and simulation kernel and user interface libraries, simulation program is generated, then import the Omnetpp.ini file and Ned files, the simulation will be running. Based on the INET framework simulation framework, this paper combines the AS6802 real-time Ethernet model provided by Core4INET to build a time-triggered Ethernet simulation system. The bottom layer (physical link) and MAC layer are all based on the standard Ethernet model in the INET framework. At the data link layer, the model in the INET framework is inherited, and a time-triggered service mechanism is added to implement the TTE protocol. This paper builds a network topology according to the simulation principle of OMNeT++. The network topology used for testing consists of 9 nodes and 3 switches, mainly record 3 types of data, as shown in Figure 4. During the simulation, the source and destination addresses of the 3 types of data will be recorded, and the latency and throughput will be recorded for performance analysis.

4.2. Simulation results and performance

The simulation time of the experiment is 15s, which compares the latency and throughput of the system with integrity assurance and the system without integrity assurance. As shown in Figure 5. and 6, after the integrity assurance is added, the latency has increased and the throughput has decreased, but it is within acceptable limits. It is proved by simulation that after the integrity assurance is added, the impact on latency and throughput is not great, but the reliability of the system is improved.
5. Conclusion

The integrity assurance is of great significance for improving the security and reliability of the TTE network. The TTE integrity assurance mechanism proposed in this paper is characterized by a combination of cryptography and have advantage in lightweight and reliable. The simulation experiment shows that, although the integrity assurance mechanism has certain influence on latency and throughput, but can effectively improve the reliability of TTE system. The next step is to add the identity authentication mechanism to further improve the security and reliability of the system.

References

[1] Kopetz H, Ademaj A, Grillinger P, et al. The time-triggered ethernet (TTE) design[C]//Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05). IEEE, 2005: 22-33.
[2] Zhang L, Goswami D, Schneider R, et al. Task-and network-level schedule co-synthesis of Ethernet-based time-triggered systems[C]//2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2014: 119-124.
[3] Bisson K. SAE AS6802 Deterministic Ethernet Network Solution[J]. Avionics Interface Technologies, 2011.
[4] Zhang Y, He F, Lu G, et al. Clock synchronization compensation of Time-Triggered Ethernet based on least squares algorithm[C]//2016 IEEE/CIC International Conference on Communications in China (ICCC Workshops). IEEE, 2016: 1-5.
[5] Wisniewski L, Schumacher M, Jasperneite J, et al. Increasing flexibility of time triggered ethernet based systems by optimal greedy scheduling approach[C]//2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA). IEEE, 2015: 1-6.
[6] Kyriakakis E, Sparsø J, Schoeberl M. Implementing time-triggered communication over a standard ethernet switch[C]//2019 Workshop on Fog Computing and the Internet of Things. Association for Computing Machinery, 2019: 21-25.
[7] Maliberan E V, Sison A M, Medina R P. A New Approach in Expanding the Hash Size of MD5[J]. International Journal of Communication Networks and Information Security, 2018, 10(2): 374-379.
[8] Jiang J, Li Y, Hong S H, et al. A Time-sensitive Networking (TSN) Simulation Model Based on OMNET++[C]//2018 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2018: 643-648.
[9] Rejeb N, Salem A K B, Saoud S B. AFDX simulation based on TTEthernet model under OMNeT++[C]//2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET). IEEE, 2017: 423-429.