2023 3rd International Conference on Electronic Information Engineering and Computer (EIECT)

AFDX network system simulation design based on OPNET

Lei Jiang *
State Nuclear Power Automation System Engineering Company. LTD
Shanghai, China
jianglei@snpas.com.cn*

Jinsong He
Shanghai Technical Institute of Electronics & Information
Shanghai, China

2023 3rd International Conference on Electronic Information Engineering and Computer (EIECT) | 979-8-3503-5770-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/EIECT60552.2023.10442638

Abstract: **Avionics Full Duplex Switched Ethernet (AFDX) has been widely used in civil aircraft avionics systems because of its advantages of low cost, delay, high reliability, real-time performance and strong scalability. The performance of AFDX network has a direct influence on the real-time data transmission, the integrity and effectiveness of information transmission of airborne avionics system. It is difficult for traditional tests to simulate on-site working conditions. Therefore, OPNET network simulation tools are used to conduct high-precision simulation and modeling of AFDX terminals and AFDX switches, and a semi-physical AFDX network system composed of simulation parts and physical parts is built as the target object of network analysis and optimization. Finally, the performance of the constructed network system is tested by means of network acquisition and monitoring, which provides an effective verification method for the performance analysis and network optimization of avionics network system. The design can improve the reliability and quality of airborne network, and play a guiding role in practical application and product design.**

Keywords-component; AFDX, OPNET, SITL

I. INTRODUCTION

With the increasing function of airborne avionics system, the requirements of avionics system for communication network performance are increasing, and the communication network needs to meet the requirements of high speed and large data volume communication [1]. Avionics Full Duplex Switched Ethernet (AFDX) has the characteristics of high reliability and high real-time, and can also meet the requirements of avionics systems for data communication. In order to improve the development efficiency of AFDX network, it is necessary to simulate and analyze the communication network at the whole machine level.

At present, among many simulation tools, OPNET is widely used in communication network modeling and simulation, using the 3-layer modeling mechanism consistent with the real network, it can realize the semi-physical simulation of complex network structure. In this paper, the network modeling and simulation tool OPNET is used to model and simulate the airborne AFDX whole machine network, analyze the communication delay of the AFDX network system, and build a semi-physical simulation test system to evaluate the AFDX simulation model, and verify the effectiveness and reliability of the technical scheme.

II. RESEARCH CONTENT

AFDX network structure mainly consists of avionics subsystem, terminal system and AFDX interconnect module.

A. Avionics subsystem

An avionics subsystem, a component of an avionics computer system that receives information from a source on an exciter, on the one hand, it is connected to a controller, sensor, or data information in an avionics system; On the other hand, it is connected to a terminal system that exclusively (uniquely) receives its data.

B. Terminal system

The terminal system is also a major component of the avionics computer system, which mainly provides a communication interface for the communication between the avionics subsystem and the AFDX interconnect module [2]. In an avionics computer system, the interface of the terminal system can provide safe and reliable data exchange to one or more avionics subsystems simultaneously.

C. AFDX interconnection module

The function of the AFDX interconnect module is to interconnect various avionics computer systems and forward AFDX frames to the appropriate destination. The module usually consists of an AFDX switch with special features. In addition, the AFDX switch also has a flow control function, which can prevent traffic anomalies caused by network failures from affecting the transmission performance of the entire AFDX network.

III. OPNET SIMULATION PLATFORM INTRODUCTION

OPNET uses Hierarchical Network Modeling, where a network model is composed of three levels: network, node and process [3]. The bottom layer is the Process layer, which uses the state machine to describe the logical behavior of nodes. The middle layer is the Node layer, which reflects the characteristics and structure of nodes, such as terminal nodes, switch nodes, etc. Nodes are composed of corresponding functional modules, such as sending and receiving frame modules, queue modules, etc. The uppermost layer is the network layer, which realizes the interconnection between nodes and corresponds to the realization of the network topology. Corresponding to the simulation model hierarchy, OPNET provides a hierarchical relationship editor. Figure 1 shows the common flow of network modeling, simulation, and analysis by OPNET. In the initial stage of modeling, an initial model is built according to the requirements, then the data to be collected is set up and simulated, and finally the collected data is analyzed. If the result does not meet the expectation, the model needs to be modified, that is, reconstructed; If the expectations are met, the simulation model can be determined.

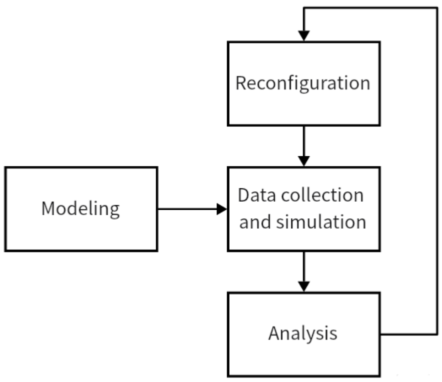

Figure 1: OPNET modeling and simulation process

OPNET provides a variety of business simulation methods, with rich collection and analysis statistics, viewing animation and debugging functions, it can directly collect the commonly used performance statistics parameters of each network level, and can easily compile and output simulation reports.

IV. SIMULATION MODELING

A. Network layer construction

The network layer is the base layer for building simulation system based on OPNET [4]. Through the OPNET project editor, the AFDX network topology is built, including the link interconnection between the terminal nodes and the switch nodes of the A and B redundant networks. The two switch typologies of the A and B redundant networks are completely disjoint, and only the terminal nodes are shared by the two redundant networks. The project editor treats terminal nodes, switch nodes, links, and so on as objects, and when an Object needs to be added to the network layer, it is done through the Object Palette component. The terminal node and switch node of AFDX network need to be defined and implemented in the OPNET node layer, and added in the project editor of the network layer.

B. Terminal node implementation

The terminal system is used to send or receive virtual link (VL) traffic. Since the corresponding relationship between the number of VL and the terminal is unknown before the simulation system loads the configuration parameters, in order to achieve the universality of the terminal model, the number of VL generating modules contained in the terminal should match the maximum number of VL sent by the terminal, and the corresponding VL traffic should be loaded by the terminal configuration table during running. The output traffic of each VL generating module of the terminal is a random model satisfying Poisson distribution, so as to achieve ergonomic transmission of each VL traffic. However, the randomly sent VL may not meet the frame interval (>=BAG) requirement, so a traffic shaping module is required to normalize the traffic of each VL. The shaped VL frame enters a FIFO queue to be sent to the output port. Taking into account redundant networks A and B, each terminal should have two sending ports and two receiving ports, corresponding to the frame sending and receiving of network A and the frame sending and receiving of network B. The received frames are fed into the redundancy management module.

C. Switch node implementation

Switch node model The number of input ports and output ports is determined. A TP control module is assigned to each input port of the switch. During operation, the TP control parameters belonging to it are loaded according to the VLID arriving at the VL. The control module records the frame loss of each VL. According to the routing configuration on the switch, the managed VL data frame that is not discarded is forwarded to the corresponding output port, and according to the VL priority parameter in the virtual link configuration table, the priority queue of the data frame should be queued in the current output port. At each priority output queue there is a probe to test the queue's backlogs based on bytes and frames, respectively.

1) Flow Control (TPA) module

Assign a TPA module to each input port of the switch, even if multiple VLS through a certain input port are not the same control module in the actual AFDX network, but due to the limitations of the physical link, these VLS cannot reach the same input port at the same time, so each VL is allowed to use the TP module for traffic control [5]. At different times, the corresponding control parameter TPConfig [VLConfig [VLID] .tpid] is loaded for the TP module by identifying the VLID that reaches the VL. OPNET implements different TP prototypes by loading different TP control parameters. The implementation principle of TP module is divided into the following steps:

a) Initialize the control parameter information of TP module: maximum credit amount:

$$TPA_i = MaxS_{max} \times \left[1 + \frac{MaxJitter}{MinBAG}\right] \quad (1)$$

credit amount recharge rate:

$$r_i = \frac{MaxS_{max}}{MinBAG} \quad (2)$$

b) Record the arrival time of the current frame t; And the arrival time t0 of the previous adjacent frame (not necessarily belonging to the same VL).

c) When the current frame arrives, the amount of available credit is:

$$credit = min\{credit + (t - t_0) \times r, TPA_i\} \quad (3)$$

d) if credit> the current frame length, the frame is forwarded;

e) else if credit< the current frame length, the frame is discarded.

f) if frame forwarding is completed, t0=t; credit=credit-current frame length;

Else t0=t.

According to the above traffic control process, each TP prototype needs to record the arrival time t0 and credit value of the previous frame. Since the routing design of the same VL in a real AFDX network does not enter from different input ports on the same switch, t0 and credit can be recorded in a single table for each switch. The array TP Record [TPNUM] [2] is used to store the values that each switch's control module needs to record during the control process. TPNUM indicates the number of all TPS in the AFDX network. Array element TP Record[i][0] stores the t0 value of the control module with TPID i, array element TP Record[i][1] stores the credit value of the control module with TPID i, and unoccupied array elements are represented by NULL.

2) Switch module

Each switch has its own routing configuration table. According to the PortID to be forwarded by the current VL, the switch module is responsible for forwarding the VL data frames that are not discarded after being controlled to the corresponding switch output port. For the VL that needs to multicast, the replication and forwarding operations need to be performed on the current switch [6]. However, since the generation time of the replicated frame is the current moment, the frame transmission time of the VL on the previous node should be added to test the end-to-end delay of the frame.

The routing table of the switch only needs to contain the forwarded PortID, and does not need to contain the switch and the input port number of the next level of connection, and the connection between the switch and the switch and the terminal and the switch depends on the link connection of the OPNET network layer.

3) Priority queue module

Each output port of the switch contains a priority queue, which contains two levels of priority. According to the priority of the VL in the virtual link configuration table, the priority of the VL frame arriving at the port is queued. Unlike the FIFO queue module, the priority queue has no module that can be called directly [7]. The implementation principle of priority queue module is divided into the following steps:

a) As long as a frame reaches the priority queue module, insert the corresponding high/low priority queue according to its priority value.

b) If the queue is in idle state, once a frame arrives, jump to c) (the initial queue state is idle).

c) If the high priority queue is not empty, the data frame in the high priority queue is forwarded; If the high priority queue is empty, go to d).

d) If the low priority queue is not empty, the data frame in the low priority queue is forwarded, and the current frame is transferred to c); If the lower priority queue is empty, the state is idle and go to b).

The priority model state machine is shown in the figure 2:

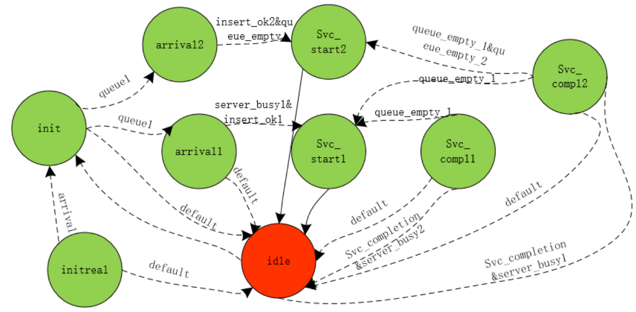

Figure 2: Priority model state machine

V. SITL SEMI-PHYSICAL SIMULATION AND DATA EXCITATION

SITL (System in the loop) is an optional add-on module provided by OPNET that allows multiple physical network interfaces to be mapped to different network addresses in a virtual network, thus making the physical device and the OPNET simulation network a unified whole [8]. It supports TCP/IP protocol cluster and is an application model specially used for network semi-physical simulation research. The simulation runs according to actual time and only needs to interact with physical devices through network cards or wireless LAN.

One of the data incentives of OPNET semi-physical simulation system is the data transmitted in the external physical object, which is converted into the OPENT module through SITL, and the other is the internal data incentive of OPNET module.

A. Data stimulation based on 664 network ICD

The data excitation based on the 664 network ICD is directly initiated by the 664 network source Tx_ComPort, and the maximum and minimum frame length, UDP, IP and other information of the frame sent by this point are constructed and sent according to the minimum time interval set by the BAG. Plus distribution (Gaussian distribution) is introduced to send randomly with the minimum interval of BAG [9].

The normal distribution probability function is:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4}$$

Use Box-Muller algorithm to obtain normally distributed random numbers:

If there are two independent random numbers U1 and U2 in the range (0,1), you can find a positive distribution of random numbers Z using either of the following equations: Z = R * cos(θ) or Z = R * sin(θ), where R = sqrt(-2 * ln(U2)),θ = 2 * π * U1 The normal value Z has an average equal to 0 and a standard deviation equal to 1, Z can be mapped to a statistic X with a mean of m and a standard deviation of sd using the following equation: X = m + (Z * sd) .

B. Data stimulation based on 664 network ICD

Fault injection refers to artificially introducing the fault into the target system according to the fault model selected in advance and using a certain strategy. By observing and analyzing the behavior of the system under the condition of fault injection, the required qualitative and quantitative evaluation results can be provided for the experimenters [10]. At the same time of fault injection, other functional modules in the network continue to send and receive information normally.

Use the program to control the node status, and automatically control the node to suddenly break or disappear frequently during the test, resulting in unstable failure of the terminal node in the network. The user can select the terminal to be manipulated through the fault injection software tool. The user can set the disconnection mode to one-time disconnection or cyclic disconnection and set the cycle period.

Build an analog terminal on the test platform, and randomly send a large number of meaningless frames from the analog terminal. The content of the data frame is automatically generated randomly by the software. The user can select the sending mode as cyclic sending and specify the sending period or select the sending at random intervals.

The utility software constructs and sends various types of data frames with application layer errors, TCP/IP protocol layer errors and physical layer errors. TCP/IP protocol layer errors mainly refer to IP address, checksum, UDP port errors, and physical layer errors mainly refer to leading code, IFG, SN, CRC, misalignment errors.

Insert an error module into the switch model, modify a certain bit of the transmitted data frame to simulate the bit flip error, modify the normal time stamp information of the frame to cause abnormal time jitter, set the credit corresponding to a VL of the switch to a value far greater than the frame length, and the simulation control function will fail. Delete the error of simulating frame filtering failure, modify the SN information of the frame to construct frame integrity loss, and modify the CRC information of the frame to construct frame verification failure.

By closing the redundancy management module in the terminal model to construct the redundancy management failure, observe whether the normal communication in the network is interfered.

VI. PERFORMANCE ANALYSIS

A. *Performance analysis calculation process*

The performance analysis tool imports data collected by OPNET to analyze network performance parameters.

The following are examples of VL delay, jitter, and dual-network deviation to describe the basic calculation process:

1) The delay of each VL in the transmission path

Record the VLi (i=1,2,...) on network A and network B respectively. Maximum values of all frame delays Di_max_A, Di_max_B;

Di_max=max(Di_max_A, Di_max_B);

2) Jitter of each VL in the transmission path

Record the VLi (i=1,2,...) on network A and network B respectively.

The maximum value Di_max_A, Di_max_B, and minimum value Di_min_A, Di_min_B of all frame delays;

Di_max=max(Di_max_A, Di_max_B);

Di_min=min(Di_min_A, Di_min_B);

Jitter= Di_max-Di_min;

3) Skewmax of each VL in A and B networks

Record the VLi (i=1,2,...) on network A and network B respectively. The maximum value Di_max_A, Di_max_B, and minimum value Di_min_A, Di_min_B of all frame delays;

Skewmax=max(Di_max_A - Di_min_B,Di_max_B - Di_min_A).

B. *Analysis result*

VL performance parameters and switch port performance parameters are used as examples. Table 1 lists VL delay and jitter data.

Table 1 VL parameter analysis table

VL parameter analysis table		
VL ID	**Maximum jitter (μs)**	**Maximum delay (μs)**
VL21	151.8	30.36
VL22	166.98	15.18
VL23	166.98	15.18
VL24	166.98	45.54
VL25	136.62	15.18
VL26	182.16	30.36
VL27	166.98	15.18

The client-to-client delay of the traditional test method is generally between 0 and 500ms, and it is difficult to simulate the network delay and jitter at the μs level. Through the simulation of this scenario, the simulation result of this model can be more close to the real application. From the above simulation results, the following conclusions can be drawn:

1) All indexes of the experimental results are in line with the data bus performance index parameters specified in AC20-156 aviation bus standard;

2) The key index parameters in the AFDX network optimized by this set of algorithms are more than double the speed of the actual engineering data;

3) It is proved that this set of algorithm model can perfectly realize the optimal design of AFDX network.

VII. CONCLUSIONS

In this paper, the OPNET simulation platform is used to build AFDX network simulation system, and the semi-physical simulation system is built. The performance test and analysis of the constructed AFDX simulation model are carried out. The test results show that the AFDX network simulation platform constructed in this paper can meet the data bus performance indicators specified in AC20-156. Thus the validity of the technical scheme is verified and the technical basis is provided for the further optimization design of AFDX network system. Preliminary verification. The design can improve the reliability and quality of airborne network, and play a guiding role in practical application and product design.

REFERENCES

[1] Holistic schedulability analysis for multipacket messages in AFDX networks. J. Javier Gutiérrez; J. Carlos Palencia; Michael González Harbour. Real-Time Systems, 2014;

[2] A Performance Analysis Framework of Time-Triggered Ethernet Using Real-Time Calculus. Xiuli Yang; Yanhong Huang; Jianqi Shi; Zongyu Cao.Electronics,2020;

[3] Yuan Wanteng, Wang Chenbo, Liu Yang. Design and implementation of AFDX Network End System Test Method [J]. Computer Programming Skills and Maintenance,2020(11):56-58.

[4] F. Ridouard, J. -L. Scharbarg and C. Fraboul, "Stochastic network calculus for end-to-end delays distribution evaluation on an avionics switched Ethernet," 2007 5th IEEE International Conference on Industrial Informatics, Vienna, Austria, 2007, pp. 559-564.

[5] Chen Xin, ZHOU Yongjun, Jiang Wenbao et al. Performance Analysis and Scheduling Algorithm of AFDX protocol [J]. Acta Electronica Sinica,2009,37(05):1000-1005.

[6] Yang Feng, Hong Yuanjia, Xia Jie et al. AFDX network technology review [J]. Journal of electronic technology applications, and 2016 (04): 4-6 + 10.

[7] Research on the scheduling method of AFDX terminal system based on time triggered and event triggered. Yuanyuan Xu; Hongjuan Ge; Jingzhong Yang. Journal of Computational Science,2017.

[8] Wang Binqi, ZHANG Yuxuan, Xue Wei. AFDX network System Simulation based on OPNET [J]. Information and Communication,2017(04):34-36.

[9] ZHAO Yongku, Wang Hongchun, Tang Laisheng. End-to-end Delay Analysis Method for AFDX Networks [J]. Electro-optics & Control,2013,20(04):81-83.

[10] L. Fernandez-Olmos, F. Burrull and P. Pavon-Marino, "Net2Plan-AFDX: An open-source tool for optimization and performance evaluation of AFDX networks," 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 2016, pp. 1-7.