Design of ARINC664 bus network test system based on WinPCap

He Jinsong[1], Zhang Yuehong[1], Hu Guosheng[1]
[1]Communication College, Shanghai Technical Institute of Electronics & Information, Shanghai 201411;
hejinsong2010@126.com

Abstract. **Aiming at the aircraft integrators ICD configuration network performance testing and analysis requirements and completing the network bus data collection and analysis tasks, an ARINC664 bus network test system based on WinpCap is designed. The hardware is designed and developed based on ordinary PC or industrial computer and high-performance Ethernet card. The software is based on WinPCap network packet capture tool for secondary development. The NDIS interface is used to filter non-ARINC664 data frames. The AFDX protocol stack is built based on WinPcap. Based on the modular design concept, the design scheme combines different functional module units to realize the design and development of the entire software. Combined with the experimental platform to verify the feasibility of design system. The system design realizes the function of analysing the performance parameters of the entire network, which is of great significance to the overall safety verification of large commercial aircraft.**

Keywords- WinPCap, ARINC664, Secondary Development, AFDX protocol stack, **modular design**

I. Introduction

As the backbone network of commercial large aircraft, the ARINC664 bus directly affects the overall safety of the aircraft. Therefore, it is particularly important to study how to improve the test technology of the ARINC664 bus network and fully verify the various performance indicators of the ARINC664 network [1][2].

At present, most of the various test software and tools used for the bus verification of the ARICN664 network have relatively single functions, low automation, particularly some commercial software is expensive, and cannot be customized for development. It is inconvenient to use and debugging. Inconvenience caused, which affects the reliability and quality of the product[3][4], so there is an urgent need to develop a platform - based and integrated application software to achieve ARINC664 bus data monitoring, network performance analysis, ICD (interface control document) analysis and packaging, EDE verification functions various functions such as testing.

II. System architecture

The hardware level of the system is designed and developed based on ordinary laptop or industrial computer and high-performance Ethernet card, which avoids the dependence on ARINC664 bus emulation card, which reduces the cost and increases the portability.

The system software is based on WinPcap network packet capture tool for secondary development. It uses the NDIS interface to filter non-ARINC664 data frames to avoid the interference of spurious frames generated by the system's own applications[5]. Using LibPcap to achieve the bottom control of the network card, you can directly operate the MAC frame to achieve the transmission and reception of Ethernet frames in the ARINC664 frame format. The interfaces used in the software development process are widely used system modules or open source tools, which can greatly improve the reliability of the software.

The software adopts the modular design idea, professionally designed according to the technical characteristics of the ARINC664 bus system, and combined design by calling different functional module units to realize the overall software design and development. The overall software architecture is shown in Figure 1.

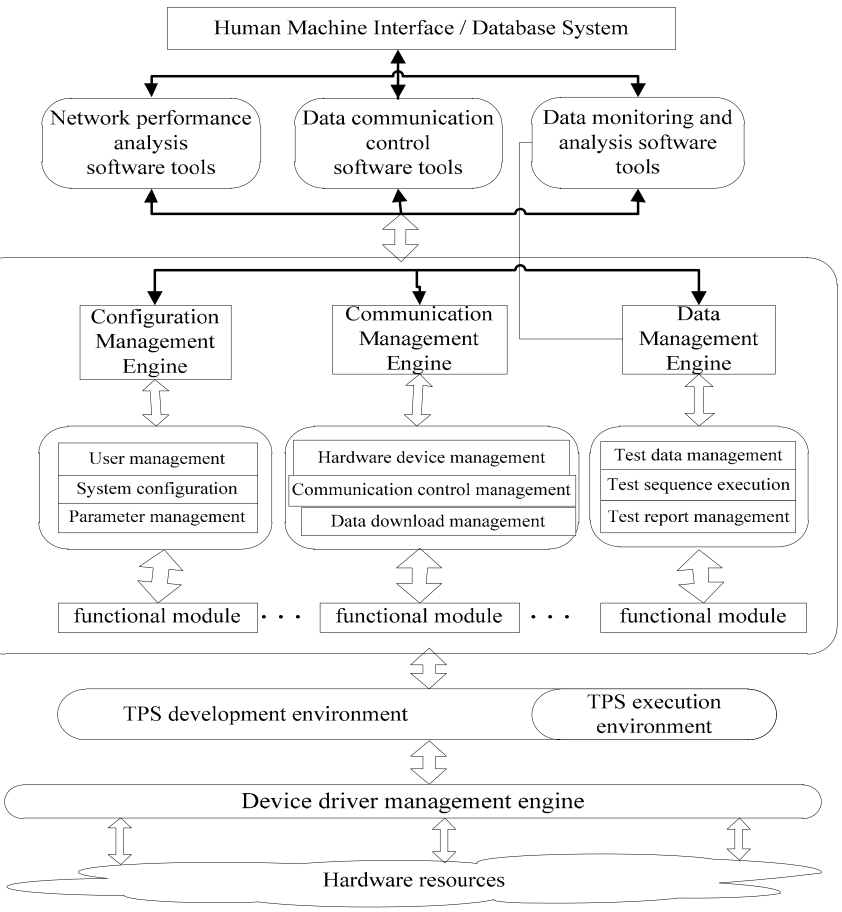
Human Machine Interface / Database System
Network performance analysis software tools
Data communication control software tools
Data monitoring and analysis software tools
Configuration Management Engine
Communication Management Engine
Data Management Engine
User management
System configuration
Parameter management
Hardware device management
Communication control management
Data download management
Test data management
Test sequence execution
Test report management
functional module
functional module
functional module
TPS development environment
TPS execution environment
Device driver management engine
Hardware resources

Figure 1. Software architecture diagram.

III. Function development

A. *WinPcap development*

The underlying Ethernet data communication control module of the application software is developed and designed based on WinPcap. The whole design idea is to encapsulate each function function provided by WinPcap into a DLL file that can be called by the host computer software, thereby completing the MAC layer of ARINC664 bus data Operation function[6][7].

WinPcap is a driver for data packet capture and analysis under the Win32 platform. It has the capture and processing of original data packets and is not processed by the operating system network protocol. In WinPcap, NPF (NetGroup Packet Filter) is the driver. Packet.dll provides a set of API interface functions after the driver[8]. Wpcap.dll provides a series of powerful packet capture interface functions for the Windows operating system. WinPcap has the functions of capturing original data packets, filtering data packets according to filtering rules, sending original data packets to fixed IP, and collecting network packet information [9].

WinPcap provides a large number of operation library functions and structures for program development of Windows systems, which involves obtaining device list information, obtaining advanced information of installed devices, opening adapters, capturing data packets, filtering data packets, analyzing and processing data packets[10], Handling offline heap files, sending data packets, and collecting statistical network traffic, etc., details are shown in Table 1. The key structures and types are shown in Table 2.

Table 1. Common functions of pcap.

Function name	Function description
Pcap_findalldevs_ex()	Get device list information
Pcap_freealldevs()	Release list information
Pcap_loop()	Open device function
Pcap_compile()	Capture function
Pcap_setfilter()	filter
Packet_handler()	Filter settings
Pcap_dump_open()	Callback
Pcap_dump()	Open heap file

Table 2. Common structures of pcap.

Structure name	Structure description
pcap_file_header	Libpcap heap file radical
pcap_pkthdr	Radicals wrapped in heap files
pcap_stat	Structure for saving statistics of an interface
pcap_if	An item in the interface list is used in pcap_findalldevs ()
pcap_addr	Represents an interface address, used in pcap_findalldevs ()

B. *Workflow design*

The basic workflow of the system software includes importing network configuration information, data reception and network testing, data analysis and record control, etc. The detailed process is shown in Figure 2.

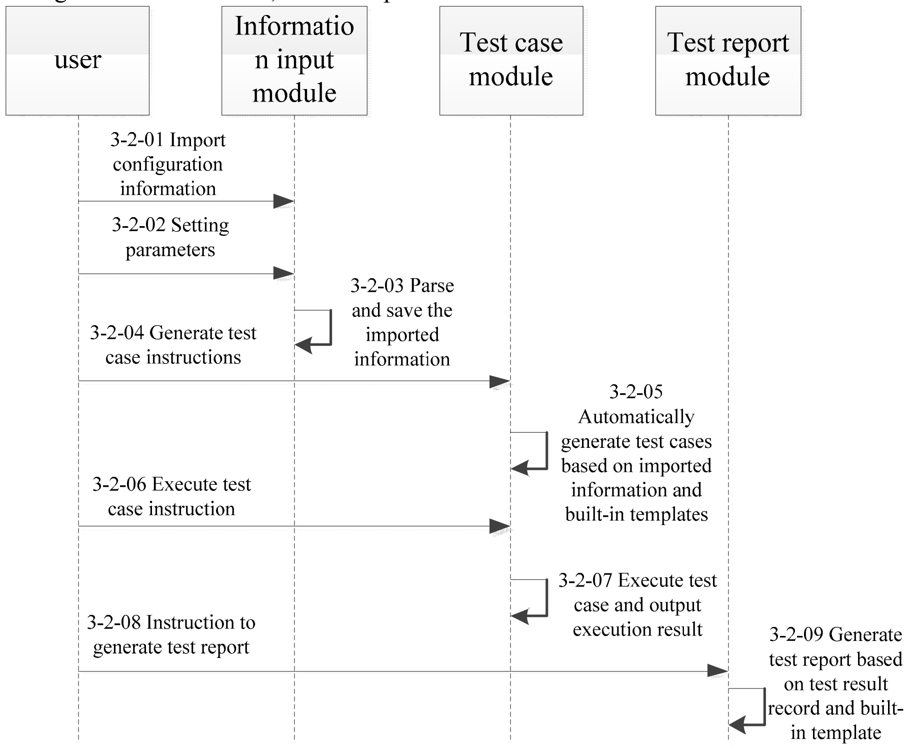
user
Information input module
Test case module
Test report module
3-2-01 Import configuration information
3-2-02 Setting parameters
3-2-03 Parse and save the imported information
3-2-04 Generate test case instructions
3-2-05 Automatically generate test cases based on imported information and built-in templates
3-2-06 Execute test case instruction
3-2-07 Execute test case and output execution result
3-2-08 Instruction to generate test report
3-2-09 Generate test report based on test result record and built-in template

Figure 2. System software workflow.

C. *Software interface development*

Interface design and development are mainly focused on applications, including management applications and test applications, design and development of test management interface, device management interface, data management interface and test application interface, providing users with simple and easy-to-operate human-computer interaction interfaces. The software provides the following main functional parts.

1) Application development, support users to develop corresponding test cases according to different ICD requirements;

2) Application management, users can manage all test cases in this function module, including rearrangement and combination, to achieve the required test function;

3) Platform management, this module provides management functions for the software itself, including resource management, component management, optimized configuration, etc .;

4) User management, this module provides information management for users.

IV. Key technologies

A. *Filtering interference frames based on NDIS*

To filter network frames using the Filter Driver, you only need to filter the transmitted frames in the FilterSendNetBufferLists () function provided by the framework. For each Ethernet frame to be sent, determine whether the first four bytes are {0x03, 0x00, 0x00 , 0x00} to determine whether it is an AFDX frame, If it satisfies the condition, then continue to call NdisFSendNetBufferLists () to send the next layer, if not, then call NdisFSendNetBufferListsComplete () to pretend that the transmission has been completed, that is, skip the next layer to send directly.

B. *Construction of AFDX protocol stack based on WinPcap*

The AFDX protocol has done a lot of customization work based on the Ethernet protocol. From the MAC layer point of view, the Sequence Number is added to the AFDX protocol to check the integrity and order of frame transmission, and the VL number is introduced to construct the destination MAC address. This means that the TCP / IP protocol interface that comes with the original system cannot be used, and you need to rebuild the complete protocol stack yourself. Using the NDIS programming interface can complete the construction of the protocol stack, but the implementation process is relatively cumbersome. WinPcap is the Windows version of LibPcap. It provides a more convenient network kernel-driven encapsulation interface. By calling these interfaces, the original data stream can be sent and received, that is, some information through its MAC layer can also be directly controlled by the application.

Use WinPcap to complete the sending and receiving of MAC frames and the automatic addition of Sequence Number. According to the ARINC 664 protocol, the Sequence Number is constantly changing based on the VL. The first frame Sequence Number on each VL is 0, and then continuously loops from 1-255. Here, by maintaining a hash table to store the Sequence Number of each VL, each time the WinPcap send function is called, the Sequence Number is added by looking up the table and adding 1. The main WinPcap library functions called pcap_findalldevs_ex () to find the network card device, pcap_open () to open the selected network card, pcap_sendpacket () to send the original data packet, pcap_next_ex () to receive the original data packet.

The generation of MAC address and the construction of frame splicing, UDP layer and IP layer are realized by writing the corresponding program modules by yourself, combined with the specific situation of the relevant protocol application in AFDX switch, the construction of the protocol stack is simplified, and the IP header is mainly completed Field padding, UDP header field padding, MAC header field padding, UDP datagram length automatic calculation, IP packet length automatic calculation, IP checksum automatic calculation and other function blocks. The information in the MAC header, IP header, and UDP header is stored using the following structure:

```
typedef struct
{
  quint8 DesMAC[6];
  quint8 SrcMAC[6];
  quint8 NetType[2];
}MACHEADER;

typedef struct
{
  quint8 VersionAndHeaderLen;
  quint8 ServiceType;
  quint8 TotalLenOfPacket[2];
  quint8 Identifier[2];
  quint8 FlagAndFragmentOffset[2];
  quint8 TimeToLive;
  quint8 HigherLayerProtocol;
  quint8 HeaderCheckSum[2];
  quint8 SrcIP[4];
  quint8 DstIP[4];
}IPHEADER;

typedef struct
{
  quint8 SrcPORT[2];
  quint8 DesPORT[2];
  quint8 TotalLenOfData[2];
  quint8 CheckSum[2];
}UDPHEADER;
```

V. Results and verification

The system supports users to perform performance testing and analysis on the aircraft's designated ICD configuration network. Combined with the actual network configuration, the performance parameters of the entire network can be analyzed according to the collected data. The WinPcap-based ARINC664 bus network monitoring system can successfully complete the network bus data collection and analysis technical requirements.

References

[1] He Zhi-qiang. Development and Important Supporting Technology of Integrated Avionics System. Telecommunication Engineering, 2004, (4): p1-5.

[2] Zhao Yong-ku，Li Zhen，Tang Lai-sheng. Research on Network Protocol of AFDX. Computer Measurement & Control, 2010, 20 (1): p8-10.

[3] Lou Xiao-qiang, TIAN Ze. Analysis and Implementation of AFDX-ES's Determinism. Computer technology and development, 2010, (8): p56-58.

[4] Liu Xiao-sheng, Wu Jin, He ye, Zhang Jie. End System Design of Aircraft Data Network Based on AFDX. Measurement & Control Technology, 2011, 30 (2):78-82.

[5] Feng Xiao-lin, Dai Wei-bing, Peng Guo-jin. Avionics bus collection and real-time analysis techniques of C919 airplane. Flight Dynamics,2016, 34 (5):73-76.

[6] Wu Tian-yi, Li Chen. Design and Development of the Data Monitoring in Cockpit Display System. Avionics Technology, 2018, 49 (2): p25-28.

[7] Wang Xiao-bing. Design and Implementation of a Network Packet Capture System. INFORMATION & COMMUNICATIONS, 2018, (2): p96-98.

[8] Hu Xiao-yuan, Shi Hao-shan, Analysis and Application of WinPcap System. Computer Engineering, 2005,31 (2): p96-98.

[9] BAI Yun, SHOU Guo-chu, HU Yi-hong, GUO Zhi-gang. Testing and Optimization Performance of WinPcap. MICROELECTRONICS& COMPUTER, 2009, 26 (7): p217-220.

[10] BAI Xue-song, Remote Network Micro Data Stacking Acquisition Simulation Based on WinPcap. Computer Simulation, 2020,37 (1): p333-337.