

Design and Research of Testing System for Vehicle Network Protocol

Anyu Cheng ^a, Xiaopin Wang ^b, Jia Li ^c, Helin Ye ^d

Chongqing University of Posts and Telecommunications, College of Automation, 400065

^achengay@cqupt.edu.cn, ^b1939547468@qq.com, ^cljorn@163.com, ^d582236889@qq.com

Abstract

There are some shortcomings in the traditional vehicle network protocol testing system. On the one hand, the local test flexibility is insufficient and the use case is difficult to be transplanted and extended. On the other hand, the whole system lacks of data support and analysis from remote server. Based on handheld wireless terminal and remote server, this paper presents a design scheme of vehicle network protocol testing system. Test cases written in XML format are automatically parsed by a handheld wireless terminal which complete the test of the vehicle network protocol through the BlueTooth communication link. The background server acquires the local test data and provides test case sets for the handheld wireless terminals. Test results show that this system is able to meet the customized testing needs of vehicle network protocols.

Keywords

Vehicle network protocol; AUTOSAR network management; database server; time delay analysis.

1. Introduction

Protocol conformance testing is an important work in the process of vehicle network development and maintain. Through twisted pair and test equipments, such as CANoe, CANview, access to the vehicle network to collect the bus message and then upload the data to the PC machine through the USB port for testing personnel to analysis. At present, the above scheme is usually adopted in testing vehicle network protocols[1-3]. In this testing process, because the connection is through the cable media, the test equipment is difficult to move and can not meet the needs of a variety of scenarios. Test cases using CAPL script language can only be used in a dedicated platform and difficult to be transplanted and expanded. Test data can not be uploaded to the remote server in real time, so it can only be analyzed on the local test equipment, lack of contrast and analysis with other test data.

Hao et al. designed a CAN-BlueTooth gateway using LPC2119 processor, which greatly improved the flexibility of the test[4]. In the scheme of Johanson et al, the packets are transmitted directly to the remote PC by Ethernet, which can not meet the needs of real-time testing[5]. Grossmann wrote test cases using the TTCN-3 standard architecture which is platform independent, and then tested them on a hybrid car[6]. This method can solve the problem that the test case is difficult to transplant. However, due to the complexity of the structure of the TTCN file and the relevance of the content, the test case is not easy to be expanded. Yang et al. introduced the remote server to the test system. Server can parse the DTC, speed and other signals from the data uploaded by the local test device[7]. Local test equipment acquisition data and then upload to a remote server, while the backend server does not provide support for local test equipment. That is to say, the communication between the local test equipment and the remote server is not bidirectional.

Based on the above analysis, this paper presents a design scheme of the test system for vehicle network protocol. The system uses the Bluetooth communication mode in literature [4], and introduces the handheld wireless terminal as the local test equipment. In order to expand easily, test cases are written in XML format. Server provides the local test equipment with a wide variety of test case sets and automatic analysis programs, which can fully extend the function of the local test equipment and meet the customized requirements of testing for vehicle network protocol.

2. System architecture

System architecture is shown in figure 1.

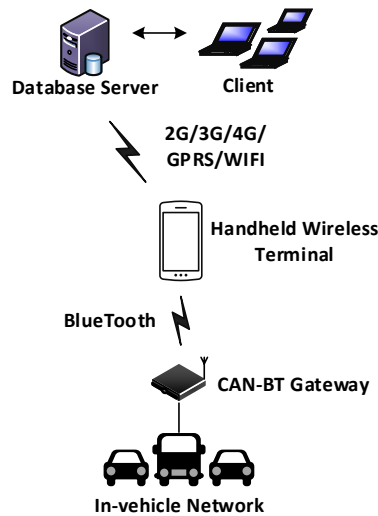


Figure 1: architecture of the test system

The vehicle network protocols in application layer are the object of the test system and implemented in the electronic control unit(ECU). CAN-BT gateway can realize the data conversion between the vehicle network data and the handheld wireless terminal. The handheld wireless terminal is equipped with the application software and test case library for vehicle network protocol testing. The user can configure the test parameters, then start testing to monitor the behavior of the ECU in the vehicle network, and complete the particular test of the vehicle network protocol. Remote server provides background data management and query services to achieve data and resource sharing. The test database established by the remote server can timely provide the field test personnel with information service. It transfers the test case set and the analytical procedure to the handheld wireless terminal and expand its function. The system's working sequence diagram is shown in Figure 2.

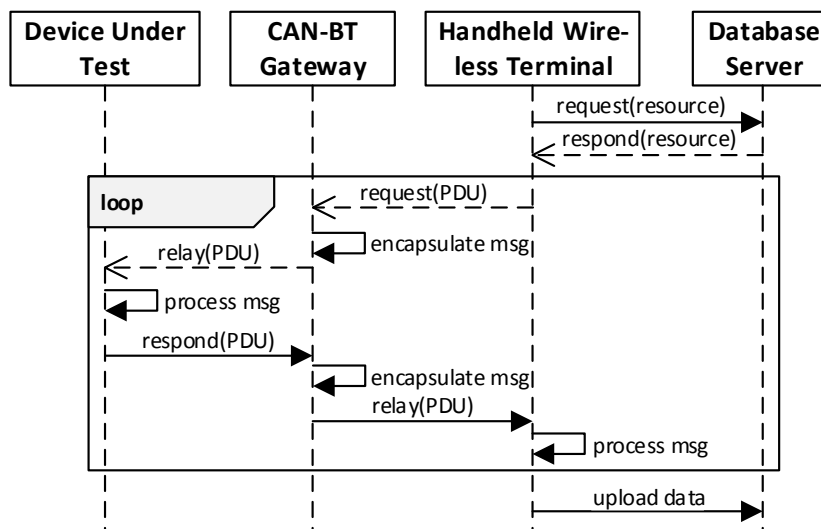


Figure 2: Working sequence diagram of the system

3. System implementation

3.1 CAN-BT gateway data conversion

In order to guarantee the integrity of data transmitted and received, this paper establishes a ring queue buffer to store the packets, as shown in Figure 3. The struct 'CAN_QUEUE' represents the ring queue

buffer, and the member variables 'front' and 'rear' respectively denote the header and the tail of the queue. When writing or reading a message in the buffer queue, the above two index numbers will change accordingly. The 'buffer' is used to store the packets, and 'size' is the number of messages in the entire queue. The value of 'lost' is used to determine if there is a packet loss in the receiving process.

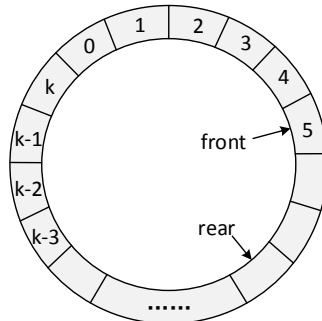


Figure 3: ring queue buffer

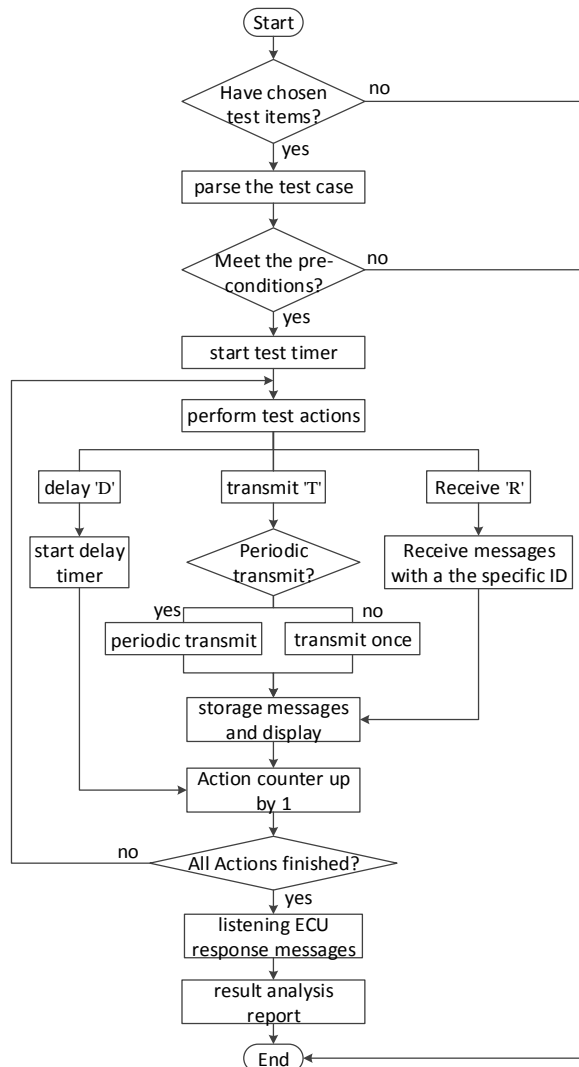


Figure 4: test case analysis flow chart

In this paper, the maximum value of the ring buffer is 50, where $k+1$ represents the actual size of the ring buffer. After the program is started, we must apply for the memory space for the ring queue buffer, and then initialize the value of the variables in the CAN_QUEUE. In order to ensure the safety of the data in the process of reading and writing, the interrupt have to be locked before the read and

write operations, and unlocked after finished the operations. At the same time, 'front' and 'rear' are also changed, indicating the entire buffer operation space.

3.2 Use case modeling and parsing

Establish test case model of AUTOSAR Network Management(AUTOSAR NM) in XML form. The 'case test' is the root element of the XML document, consisting of three attributes and three subelements. Title, number and version of the test case are respectively described by 'name', 'num' and 'version'. The state of the test node before the execution of this test case is given by 'precondition'. The whole test execution process is abstracted into three kinds of actions, namely, receive, transmit and delay. Receive or delay action only contains one parameter. The sending action contains six parameters, including ID, mode, period, source node address, Control Bit Vector(CBV), and custom data. Timers and CBV value can be used to determine the 'expected result', which is the desired state when the test case is executed.

Network management protocol testing process is the essential content of test case analysis, as illustrated in Figure 4. After the user has configured the parameters and started the test, the Android application calls the corresponding functions to obtain the ID of the test case. Program goes through the XML file to find the pre-condition, test steps and the expected results and then get the information stored in a predefined data structure. Subsequently, the system performs the actions in accordance with the test procedure which contains multiple attributes. Actions are divided into three categories. 'R' means to receive an expected message, 'T' means to send a particular message, 'D' means the delay time. Depending on the type of action, the test software responds accordingly.

3.3 Server build

In this paper, we build a database server, whose main function is to receive and store the data uploaded by the handheld wireless terminal. At the same time, handheld wireless terminals and remote clients can also be connected to the server to read data from the database. Because the server needs to receive and store large amounts of data from the mobile terminals, we choose the SSH framework to build the server which is convenient for the update and maintenance. SSH is an integrated framework of struts+spring+hibernate, which is a popular Web application framework currently[9]. The process of uploading data to a remote server with a wireless terminal is shown in Figure 5.

After received the request from the mobile terminal, the server passes the information to the CanMessageAction class. In the CanMessageAction class, the information is encapsulated into an CanMessage entity class. Then the entity class is passed to the business logic layer by means of add() function in CanMessageAction. Then the business logic layer calls the save() function in the persistence layer to save the entity class to the database.

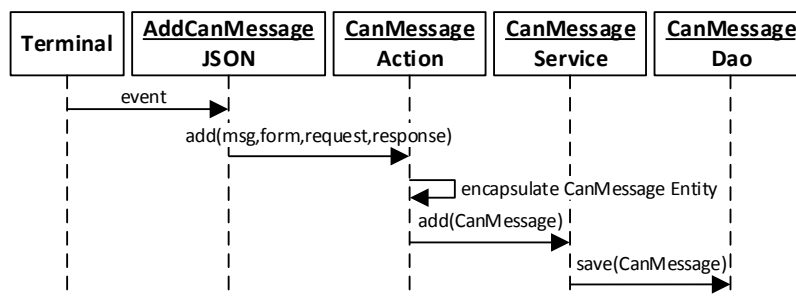


Figure 5: Data upload sequence diagram

4. System testing and data analysis

4.1 System functional test

In this paper, the functional verification of the test system is carried out in a vehicle instrument which has realized the function of AUTOSAR network management. Test procedure is as follows:

- (1) Connect the instrument and CAN-BT gateway to vehicle network, and then supply power.

- (2) Open the test software in the smart phone, configure the wireless network, find and connect the Bluetooth gateway.
- (3) Select the 'AUTOSAR NM' test case set in server tab and download, as shown in Figure 6(a).
- (4) Select 'RM to NO' in test tab, and click the start test button. The bargraph shows the progress of the test in real time, as shown in Figure 6(b).
- (5) The real-time interactive process of the message can be viewed in the 'view' tab, as shown in Figure 6(c).

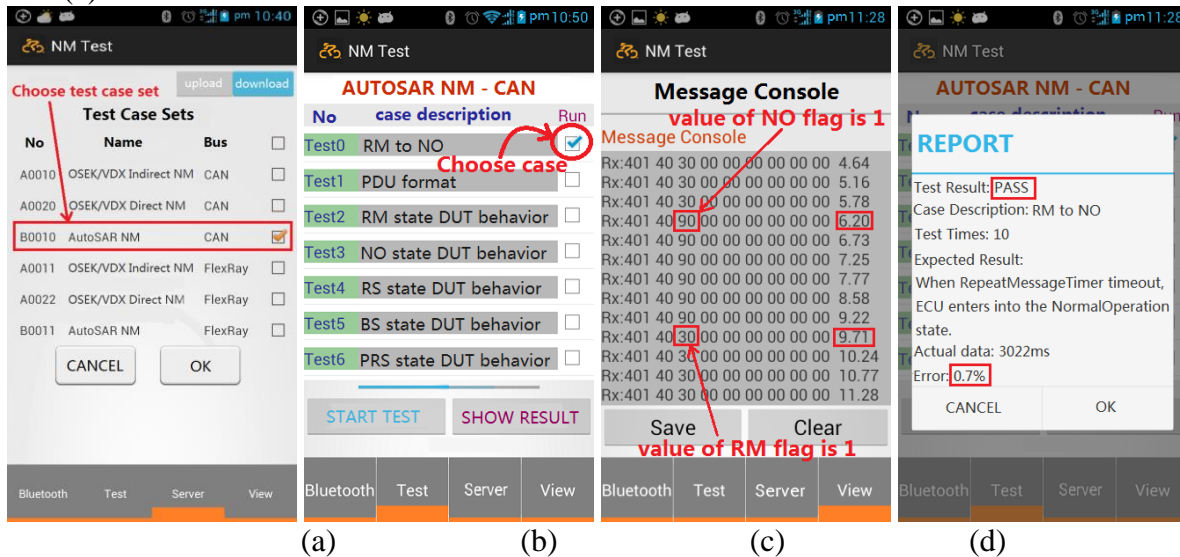


Figure 6: The process of executing a test case

From the 'view' interface, the instrument sent a network management message of which the CBV is 0x90 in 6.20s, which indicates that the instrument is in the RepeatMessage state. The CBV value of the network management message is changed to 0x30 after 3.02s. The state of instrument has switched to NormalOperation. So the time parameter of RepeatMessageTimer is 3.02s. The error rate is $(3.02-3) / 3 = 0.7\% < 10\%$, within the prescribed error range. Based on the above analysis, the case is successfully passed, and its report is shown in Figure 6(d).

Similarly, we tested all the remaining 27 NM AUTOSAR cases designed in this paper. Test cases are divided into 7 categories: initial test(1), PDU format tests(2), error detection and recovery tests(3), state tests(5), state transition tests(9), timer parameter tests(4) and sleep or wakeup tests(4). Test results of state transition test are shown in Table 1.

Table 1: AUTOSAR NM state transition testing

No	Test case	CBV	Timer	Result
1	BS to RM	none to 0x90	MCT=500ms	pass
2	RM to NO	0x90 to 0x30	MCT=494ms, RMT=3020ms	pass
3	NO to RS	0x30 to 0x18	MCT=511ms	pass
4	RS to RM	0x18 to 0x90	MCT=521ms	pass
5	RM to RS	0x90 to 0x18	MCT=499ms, RMT=2940ms	pass
6	RS to NO	0x18 to 0x30	MCT=501ms	pass
7	NO to RM	0x30 to 0x90	MCT=500ms	pass
8	RS to PBS	0x18 to none	MCT=507ms	pass
9	PBS to BS	-	MCT=515ms, BST=3070ms	pass

Note: BS (BusSleep), RM (RepeatMessage), RS (ReadySleep), NO (NormalOperation), PBS (PrepareBusSleep), BST (BusSleepTimer), RMT(RepeatMessageTimer), MCT (MessageCycleTimer)

4.2 Communication link analysis and test

A typical message transmission delay T_{delay_t} for vehicle bus is defined as the time interval between the time T_s that the source node begins to transmit the message and the time T_d that the destination node receives it completely. It consists of 3 parts: the packet transmission delay of the source node T_{sst} , the transmission delay in the network channel T_{prop_can} and the target node's receiving delay T_{drv} . In this paper, the gateway is used as a relay node, which adds the gateway processing delay T_{deal} and Bluetooth transmission delay T_{prop_bt} . The transmission delay model is shown in Figure 7.

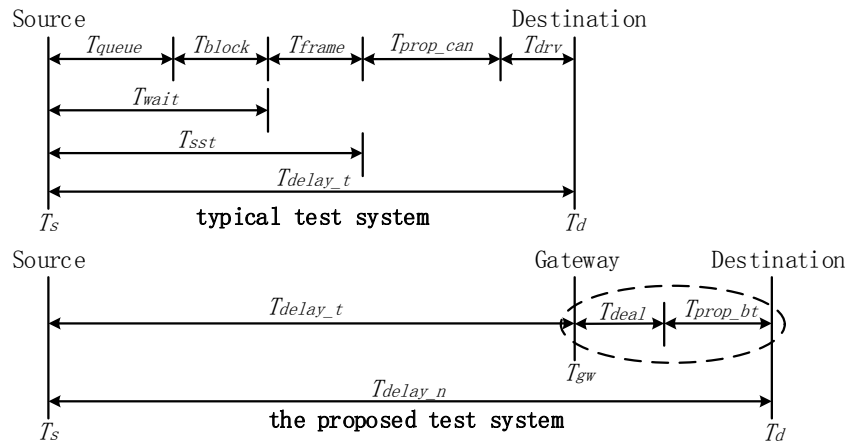


Figure 7: transmission delay model

In this test system, the operating frequency of the CAN-BT gateway processor is 72MHz. There are 11 effective bytes in a CAN message and 14 bytes in a BlueTooth packet. The data bus width of the ARM processor is 32 bits. Read or write a message needs to consume 7 clock cycles. In normal circumstances, data format conversion takes up 1000 clock cycles at most. A packet processing delay is

$$T_{deal} = 1000 / 72MHz = 13.9us$$

In Bluetooth transmission, the random variable \bar{N} is defined as the average number of packets to be transmitted if it is successfully transmitted. Assuming that the transmission of a message is failed continuously in previous $n-1$ times and successful in the next time, the average times is

$$\begin{aligned} \bar{N} &= 1 \cdot (1 - P(\gamma_1) \cdot (1 - P_r(\gamma_2))) \\ &+ 3 \cdot P_r(\gamma_1) \cdot P_r(\gamma_2) \cdot (1 - P_r(\gamma_3)) + \dots \\ &= \sum_{i=1}^{\infty} \left(i \cdot (1 - P_r(\gamma_i)) \cdot \prod_{n=1}^{i-1} P_r(\gamma_n) \right) \end{aligned} \tag{1}$$

Among them, γ_i is the instantaneous SNR in the process of sending or returning the i -th packet. Because of the assumption that a packet is not related to other packets, namely the Quasi-static fading, all packets have the same probability density function of SNR(Reference 8). therefore

$$\bar{N} = \sum_{i=1}^{\infty} i \cdot (1 - \bar{P}_r(\bar{\gamma})) (\bar{P}_r(\bar{\gamma}))^{i-1} \tag{2}$$

We use $\bar{P}_r(\bar{\gamma})$ to represent the corresponding average retransmission probability. Because the average SNR $\bar{\gamma} = E\{\gamma\}$, then it can be further calculated that

$$\bar{N} = \frac{1}{1 - \bar{P}_r(\bar{\gamma})} \tag{3}$$

For any type of packet, the average throughput R is a function of the average retransmission number \bar{N} . So

$$R = \frac{K}{D\bar{N}625 \times 10^{-6}} = \frac{K(1 - \bar{P}_r(\bar{\gamma}))}{0.000625D} \tag{4}$$

In the formula, D is the number of slots occupied by each transmission, including the return packet. The length of the valid data in the packet is represented by K .

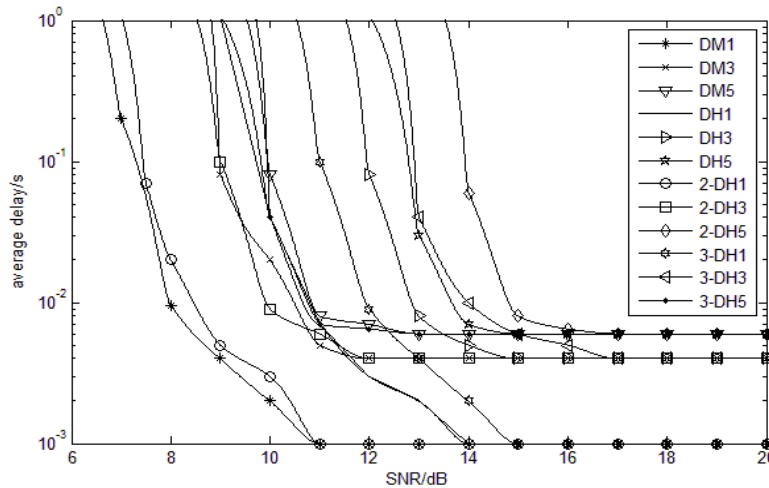


Figure 8: Average delay of all data packets

According to the formula (4), the average transmission delay can be expressed as

$$\bar{t}_d = 0.000625(D\bar{N}) \tag{5}$$

Figure 8 shows the theoretical value of the average delay for each type of packet.

In the laboratory environment, we test the time delay caused by the gateway through an oscilloscope. The results are described in Figure 9.

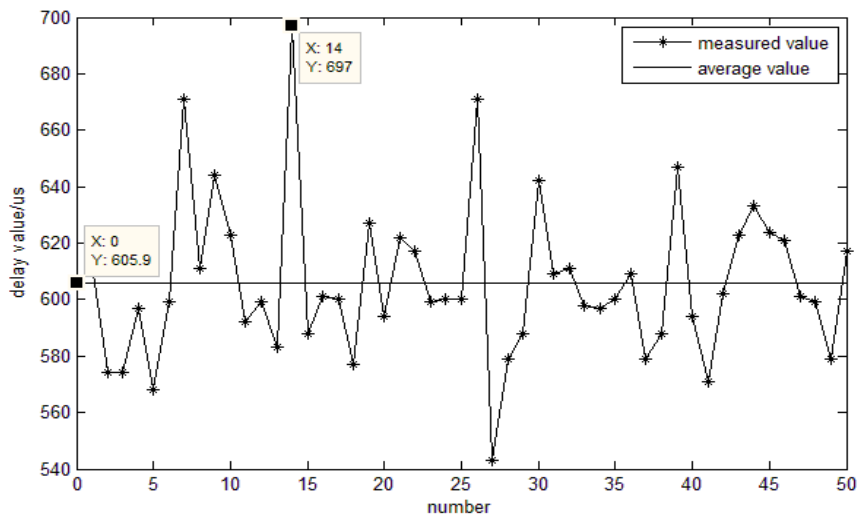


Figure 9: Time delay caused by gateway

The peak value of the gateway delay is 697us, while the average value is 605.9us. By setting the timer, the network management system can judge the state of the whole network by monitoring the sending and receiving time of NM messages. If the delay exceeds the error range of the value of the timer, the probability of the error of the state estimation is greatly increased, which means the test system

designed in this paper is worthless. Therefore, it is necessary to calculate the ratio between the gateway delay and the standard value of NM AUTOSAR timer. The results are shown in Table 2.

According to the experimental results, all the time delay parameters introduced by gateway are not beyond the error range of the standard NM AUTOSAR timers. T_WAKEUP_WAIT, T_WAKEUP_SYNC, T_SLEEP_WAIT, T_KEEP_NORMAL_MIN_TIMEOUT and T_ACTIVETX are related to the packet interaction process, which determines the status of network management. The delay error caused by the gateway is less than 1%, which has a very low impact on the results. T_START_WAKEUPTX, T_WAKEUPTX and T_START_ACTIVETX specifies the time interval for a single node to transmit the corresponding message after the state is switched. Although the delay error caused by the gateway is higher than the former, but it does not exceed the scope of the error and has no effect on the protocol test results. Therefore, the test system designed in this paper can meet the needs of real-time testing of vehicle network protocols.

Table 2: time delay of NM AUTOSAR timers

Timer name	Value	Tolerance	Delay ratio
T_WAKEUP_WAIT	650ms	+/-10%	0.107%
T_WAKEUP_SYNC	100ms	+/-10%	0.697%
T_SLEEP_WAIT	2000ms	+/-10%	0.035%
	3000ms	+/-10%	0.023%
T_KEEP_NORMAL_MIN_TIMEOUT	3000ms	+/-10%	0.023%
T_START_WAKEUPTX	10ms	+/-10%	6.970%
T_WAKEUPTX	20ms	+/-10%	3.485%
T_START_ACTIVETX	10ms	+/-10%	6.970%
T_ACTIVETX	500ms	+/-10%	0.139%

5. Conclusion

Although the scheme proposed in this paper can meet the basic needs of the vehicle network protocol testing, it still has many shortcomings. In this system, all the packets are transported in plaintext by CAN-BT gateway and could be sniffed or intercepted easily. So it is necessary to consider the security issues in the process of data communication. In addition, the system is tested only in the laboratory environment. Due to the fact that the system has not been tested in the electromagnetic environment, the anti-jamming performance of the system is unknown in the practical application environment. Overall, the testing system designed by this paper has a certain engineering application value, and has a reference significance for the development of the vehicle network protocol testing system.

References

- [1] Rishvanth, Valli D, Ganesan K, et al. Design of an In-Vehicle Network (Using LIN, CAN and FlexRay), Gateway and its Diagnostics Using Vector CANoe[J]. American Journal of Signal Processing, 2011, 1(2):40-45.
- [2] Kucera P, Hyncica O, Honzik P, et al. On vehicle data acquisition system[C]// Conference Record - IEEE Conference on Intelligent Transportation Systems. 2011:1446-1451.
- [3] C.Y. Yu, D.X. P, Z.J. L. Vehicle CAN bus data recorder based on MSP430, Foreign Electronic Measurement Technology, Vol. 30 (2011) No. 2, p.35-37.
- [4] Y.S. Hao, X.L. Zhao, B. Zhu. Bluetooth CAN Bus Analyzer Design Based on ARM, Mechatronics, Vol. 20 (2014) No. 04, p.57-60.

-
- [5] Johanson M, Karlsson L, Risch T. Relaying Controller Area Network Frames over Wireless Internetworks for Automotive Testing Applications[C]// Systems and Networks Communication, International Conference on. IEEE, 2009:1-5.
- [6] Grossmann J. Testing hybrid systems with TTCN-3 embedded[J]. International Journal on Software Tools for Technology Transfer, 2013, 16(3):1-21.
- [7] Yang Y, Chen B, Su L, et al. Research and Development of Hybrid Electric Vehicles CAN-Bus Data Monitor and Diagnostic System through OBD-II and Android-Based Smartphones[J]. Advances in Mechanical Engineering, 2015, 5(3):741240-741240.
- [8] D. Eastlake 3, Reagle J, Solo D. (Extensible Markup Language) XML-Signature Syntax and Processing[J]. Java™ Platform Revealed, 2002, 18(1):115-146.
- [9] Johnson R. J2EE Development Frameworks[J]. Computer, 2005, 38(1):107-110.
- [10] MATTHEW C. On the throughput of Bluetooth data transmissions[C]//IEEE Wireless Communications and Networking Conference. Orlando, Florida: IEEE, 2002: 119-123.