

Vehicle - based Heterogeneous Network Interconnection Model Based on Virtual Device

YIN XiaoHong, He Miao

Nanchang Institute Of Science & Technology, Nanchang 330108, China

Key words: Vehicle gateway; virtual device; Linux system; heterogeneous network

Abstract: In order to solve the problem of the communication between the heterogeneous network and the various types of equipment and the Internet, a new, Vehicle heterogeneous network interconnection model based on Linux virtual device. In this model, the dynamic load characteristics of Linux modules and the virtual network equipment, To achieve the unification and interconnection between different types of devices Logically, A virtual network device layer is designed between the TCP/IP protocol stack and the device driver layer, All data in the system kernel and the real device communication through the virtual network device layer, The virtual network device provides a unified socket layer interface, The design and implementation of the upper application service is more simple .Experimental tests show, The model can be used to communicate between heterogeneous networks, Especially with Internet communication.

Introduction

With the rapid development of mobile communication and vehicle electronic technology, The combination of the two is more and more closely, There are various kinds of heterogeneity in the current vehicle network. Network, due to hardware characteristics, Transmission format and other aspects of the different, Lead to difficulties in communication, and with the development of a variety of mobile application services, It is more and more urgent to get the information of vehicle network through internet, The key equipment to solve this problem is the vehicle gateway. Vehicle gateway in literature [1]. Application layer transmission technology, All data processing is done at the application layer, Consider the appropriate hardware characteristics. For application development and. There is a certain degree of difficulty, At the same time, the requirements of the vehicle gateway can be effective. Distribution of various network data, As far as possible online ability, High throughput, high maintainability, etc., Therefore, this paper constructs a kind of heterogeneous network interconnection model based on virtual equipment, Improve the performance of the vehicle gateway.

Existing technology analysis

The existing common vehicular network includes control area network CAN (con-)Troller area network),Local interconnect network LIN (local interconnect)Network) etal.CAN is a German Bosch Company at the beginning of 1980s (Bosch),In order to solve the problem of the data exchange between the control and test instrument in the modern automobile, A serial communication network which can effectively support distributed control or real-time control is developed.LIN is a low cost serial communication network. Distributed electronic system control for automotive applications, and the vehicle has a common electronic equipment GPS receiver [4-5], FM radio and sensor [6] etal. The vehicle gateway can effectively combine the information data of these devices or networks, And provided to the monitoring terminal. The existing technology is to achieve the integration of various data through the application layer, And through the CD-MA wireless network will be sent to the remote monitoring terminal, Or through other connections to the local query analysis, Such as OBD Obstacle detector.

Internet is the most widely used internet, Vehicle information communication via wireless network has been a hot spot of automotive information application A.For example, in 1998 the

company (Clarion) and Microsoft cooperation in the development of the car's computer system "music PC, Japan pioneer and Sony Corp jointly launched with laser disc entertainment function of the GPS positioning system. These systems will only be limited to the vehicle network or vehicle electronic equipment, Lack of complete interconnection system, This is related to the heterogeneous complexity of the vehicular network and the difficulty of application design in heterogeneous networks. Linux virtual devices can effectively simulate some real devices. Work effect, Through the design of virtual network equipment to unify the heterogeneous network and electronic nodes, Can effectively shield the differences between heterogeneous networks and electronic nodes, And provides a consistent application programming interface for the upper application, So as to reduce the design difficulty of vehicle communication program, Increase program portability. Sex and maintainability.

Model establishment

System model overview. The basic idea of the model is, When accessing a network node or an electronic device via CDMA or WLAN, All data packets are treated as IP network messages and follow the network security policies and methods. And from the vehicle network nodes or electronic devices to send data out through TCP/IP. Protocol stack for data interaction. Thus, the application of the upper layer covers the characteristics of various electronic node hardware. This model unified the application layer transmission interface and programming interface, It makes it easier to design and write the application layer service program. At the same time, the module is designed to support IPv4 and IPv6 dual stack mode. The schematic diagram of the model is shown in Figure 1.

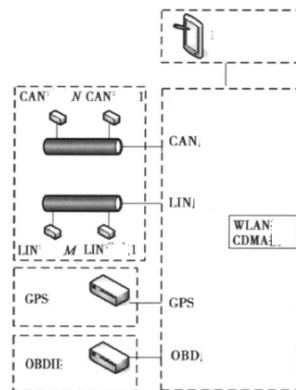


Fig.1 The structure of the model

Gateway connects the remote central server via WLAN or CDMA, CAN virtual interface that CAN virtual device interface to connect to the CAN network; LIN virtual interface that LIN virtual device interface to connect to the LIN network; GPS and OBD virtual interface are connected to the GPS receiver and OBDII devices. Each virtual device within the gateway will maintain a IPv6 address for the corresponding vehicle control network or electronic node. The protocol between heterogeneous networks is transformed in the management module of each virtual network device. Data obtained from the corresponding hardware drivers of the virtual network device module, The corresponding management module is used to analyze the format. Then, the virtual network device module unified the data into the network data structure by buff, so as to realize the unification of data format, Therefore, the upper application data is a unified IP message. 3.2 component analysis of vehicle gateway model. The whole system model. The most important is the component model of the vehicle gateway.

All of the electronic nodes and vehicles in the network devices are virtualized as one of the Ethernet devices, So in addition to the LAN device, Other network devices have virtual devices for access management. The virtual network device interface layer runs logically on the relevant real device driver layer. All of the read and write operations of the real device are realized by the virtual network device, and the realization of the virtual network device is based on the Linux virtual device. Work and management is not the same as ordinary Ethernet card.

mathematical analysis of performance. According to the characteristics of each virtual network equipment module, The performance of any virtual network device can be considered using the M/M/1 model in the queuing model. Among them, the definition of the customer for each remote monitoring terminal or local users at any time to request the vehicle gateway forwarding the corresponding data packets; The process of forwarding packets in response to a service gateway virtual network device; The server is a virtual network device for a vehicle gateway. Taking the CAN virtual network device as an example, due to the randomness of task arrival, The service provided by CAN virtual network device to each task is not uniform. So it can be assumed that the customer arrival process is Poisson distribution, The service hours of the attendants are exponentially distributed. But in the actual test, the vehicle CAN node is the same node, There is no priority problem, Therefore, the same level of task queue rules for FIFO (FIFO), Therefore, the queuing model can be treated as a single level, As shown in Figure 2.

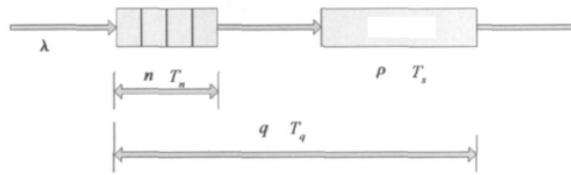


Fig.2 M/M/1 queueing model

The average arrival rate; N wait for service. The number of data nodes in the queue; T_n wait time for task; T_s for the server (the CAN virtual network equipment) for a single task. Business Hours; P waiter utilization; Q represents the total number of tasks; T_q the time required for each task to be consumed by the system, including waiting time and service time. At the same time as the definition of the number of services per unit time is dealt with $1/E(T_s)$, $E(T_s)$ represents the average service time of the task, thereby. Get $P = \lambda / \mu$, according to the actual significance, it should meet $0 < P < 1$. = P In order to describe the performance of CAN virtual network devices, average. The queue length $E(n)$ and the mean sojourn time of tasks $E(T_q)$. $E(n)$ The average number of queues waiting for the CAN virtual network device to be processed. In the case of finite queue length, the system can be effectively reflected. Possibility of birth cohort blocking; $E(T_q)$ is the CAN virtual network device model. The average time of block processing tasks reflects the CAN virtual network. Equipment handling capacity. From these two values, we can approximately reflect the average ability of the system to deal with a certain data message.

The λ of type (1) indicates the average arrival rate of packets, $1/E(T_s)$ to deal with the number of packets per unit time CAN virtual network device. In order to get P , namely the use of CAN virtual network device rate; Type (2) - (1) according to the p , Computing CAN virtual network. Average queue length $E(n)$; Type (3) the average number of packets and arrival rate $\lambda \tau$ unit through the CAN virtual network device can process the data packet, The average time $E(T_q)$ for processing tasks of CAN virtual network device modules; Formula (4) $E(T_q)$ calculated from the average arrival rate of the packet and the formula (3), Gets the total number of packets arriving. In this system, we can estimate the approximate performance of the system by using λ and T_s Implementation of 3.4 CAN virtual network device Specific implementation to CAN virtual network equipment as an example to illustrate. The principle of virtual network device reference Ethernet device driver, The dynamic loading mechanism of Linux is added to the kernel of Linux system dynamically.

In order to connect with the heterogeneous network vehicle electronic equipment and network node, The network device media layer is driven by the original equipment, Device driver function layer that provides actual functionality, The main function is to realize the data transmission between the network device interface layer and the corresponding device driver. The system is implemented by the virtual network device through the kernel module, At the same time have the ability to access management. Access management is responsible for determining whether the data needs to be forwarded to the local user to monitor the IP address, Otherwise, the data sent to the CAN through the can write device driver to handle The second layer network device interface layer

is simulated by software, The top layer protocol interface layer is exactly the same as the Ethernet driver, In order to realize the virtualization of heterogeneous network vehicle electronic equipment, Then the application layer interface is unified, Unified use of socket API to program design and Implementation. There are 2 main forms of Linux network device driver: One is through the kernel to load; the other is the module loading. In this paper, the virtual network equipment is designed in the form of module loading.

The module can insert the virtual network device driver into the kernel by loading the command insmod, Insmod will call module.in.it (can)Moduleinit) function, The initialization function pointer initialization of virtual network device is initialized, That is, the module init function will call can module The init function, in the can module init function will call registernet-dev (Can control) registration function of the network equipment in the system, Then initialize the device to complete the loading of the device. Insert the device data structure to the end of the debase list. If necessary, you can uninstall the virtual network device driver module by calling the module exit function in the network driver by executing the module.

Test

Experimental Environment. The module can insert the virtual network device driver into the kernel by loading the command insmod, Insmod will call module init (can)Module nit) function, The initialization function pointer initialization of virtual network device is initialized, That is, the module init function will call can module. The init function, in the can module in it function will call register net-dev (Can control) registration function of the network equipment in the system, Then initialize the device to complete the loading of the device. Insert the device data structure to the end of the debase list. If necessary, you can uninstall the virtual network device driver module by calling the module.exit () function in the network driver by executing the module rmmmod4 test4.1 experimental environment.

Taking CAN node as an example to test, Where the vehicle gateway hardware using ARM9 processor as the core, Peripheral interface includes 1 CAN interface, the RS232 interface and the 4 USB interface, The operating system uses the ARM-Linux kernel version number 2.4.18-rmk7 Remote test server using CPU as PIII800, Memory PC 1GB, The operating system is RH9, kernel version 2.4.20-8. The CAN core of the test is made by PIC18F2580 chip. Self designed node internal program, Only use LED to display the state of the instruction. The main content of the test is to test the performance of the Unicom test and virtual device module. Test program using IPv6 socket. API[13-14] to write, Transport layer protocol using UDP[15]. IPv4toIPv6 proxy server is responsible for IPv4 protocol and IPv6 Conversion between protocols The structure of the test system is shown in Figure 3.

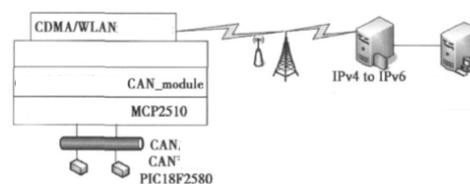


Fig.3 Structure of the test system

The vehicle gateway is in the CDMA/WLAN wireless network communication module and CAN virtual network device, CAN virtual network device through the CAN device driver and CAN bus communication, The external network is connected to the remote server through the CDMA/WLAN module. 4.2 test and evaluation. Connectivity test, Single node access test, Sends the designed instruction from the remote test server, Observe whether the node receives the instruction and whether the return instruction is accurate; Multi node test, Observe the operating conditions under the condition of multiple instructions. Each test sends instruction interval 2s, Test 8h, record the number of times the command failed to send The results are shown in Table 1

Table 1 test results of Unicom

Test times	Failure times
1	1
2	0
3	1
4	0
5	0

Virtual device module run time performance test, Mainly involves 2 kinds of feelings. A total of 4 time points. The first case is that the data needs to be forwarded to the user, The time to record the data arrival module T1, Module for IP filtering, Conversion protocol and forwarded to the user to monitor the time T2, The difference between T2 and T1 is the time consumed by the module to process and forward the data T1; The second case is the data directly sent to the vehicle through the module CAN electronic node, The time to record the data arrival module T3, The data is then forwarded to the CAN node and the results returned by the CAN node, After receiving the return result, the module is encapsulated into the IP data packet into the protocol stack, At this time the record time T4, T4 and T3 difference of T34 for the total time of data processing. Each test packet transmission interval 0.5s, Each group sends test message 600 (about 5min), recording the mean. The test results are shown in Table 2.

In fact, In practical application, For CAN node independent access to the instruction sent, It is carried out by the manual remote monitoring terminal or the local human computer interaction interface, So the interval is set to 0.5s is the limit state.

Table 2 time efficiency test

Group number	T12/ms	T34/ms
1	44.02	382.02
2	44.08	382.08
3	45.13	382.18
4	44.81	381.59
5	44.29	381.87

From the above data can be seen, The model can be used to transmit data in heterogeneous networks, And through the re packaging of a variety of vehicle electronic equipment driver, The interface of the application layer is unified into a virtual network device interface, Make the program development and maintenance more convenient.

Conclusion

In this paper, the design of the heterogeneous network interconnection model based on virtual network equipment, The module loading characteristics of Linux are effectively utilized, Thus, a variety of electronic devices and heterogeneous network nodes are re packaged, Create a virtual network interface layer, Build a system for different applications. An interface platform. This design enables the design of the application, Development and maintenance is more simple, It can provide a broader development environment for future vehicle applications. At the same time, the virtual network interface layer is in the driver layer, So it has nothing to do with the IP protocol, Can support IPv4 and IPv6 dual stack, And most of the security policy based on Internet network can be obtained. Effective support, For the future expansion of the application to provide a wider range of support. The next step is to consider how to improve the efficiency of each virtual device, And a more reasonable queue scheduling method.

Acknowledgements

Nanchang key laboratory and characteristic specialty(NGTSZY201001).

The science research project of the jiangxi provincial education department(GJJ151239).

The science research project of the jiangxi provincial education department(GJJ151235).

References

- [1] Hao B, Liu Y, Wei D, et al. Research on the Interconnection Model between Vehicular CAN

- Network and Internet Based on In-vehicle Gateway[C]// NETWORKING and Parallel/distributed Computing. New York: IEEE Press, 2012:pp.615-620.
- [2] Hao B, Liu Y, Wei D, et al. Research on the Interconnection Model between Vehicular CAN Network and Internet Based on In-vehicle Gateway[M]. 2012.
- [3] Yuan L. A heterogeneous wireless network interconnection strategy based on IP switching[C]// Computer Science and Automation Engineering (CSAE), New York: IEEE Press., 2011:pp474-477.
- [4] Javadi B, Abawajy J H, Akbari M K. Analytical modeling of interconnection networks in heterogeneous multi-cluster systems[J]. The Journal of Supercomputing, 2007, 40(1):29-47.
- [5] Yin J, Zhou P, Holey A, et al. Energy-efficient non-minimal path on-chip interconnection network for heterogeneous systems[C]// Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design. New York: ACM Press., 2012:pp57-62.
- [6] Zheng J, Wu Z, Hu M, et al. Communication controller for heterogeneous network interconnection[J]. Automation of Electric Power Systems, 2003.
- [7] Bhansali S, Chapman G, Jain V. Inter-layer vias and TESH interconnection network for a 3-D heterogeneous sensor system on a chip[J]. Proceedings of SPIE , 2005, 5796(39):6224-6231.
- [8] Flores A, Aragón J L, Acacio M E. Efficient Message Management in Tiled CMP Architectures Using a Heterogeneous Interconnection Network[J]. High Performance Computing – Hipc, 2007, 4873:133-146.
- [9] Lodde M, Roca T, Flich J. Heterogeneous network design for effective support of invalidation-based coherency protocols[C]// Interconnection Network Architecture. New York: ACM Press, 2012:pp1-4.
- [10] Ilavarasan E, Thambidurai P, Punithavathi N. Task Scheduling Algorithm for Interconnection Constrained Network of Heterogeneous Processors[M]// Intelligent Information Technology. Springer Berlin Heidelberg, 2004:31-39.
- [11] Kuorilehto M, Hännikäinen M, Hämäläinen T D. A Survey of Application Distribution in Wireless Sensor Networks[J]. EURASIP Journal on Wireless Communications and Networking, 2005, 2005(5):859712.
- [12] Louati W, Zeghlache D. Network-based virtual personal overlay networks using programmable virtual routers[J]. IEEE Communications Magazine, 2005, 43(8):86-94.
- [13] Wu J, Wang Y. Hypercube-Based Multipath Social Feature Routing in Human Contact Networks[J]. IEEE Transactions on Computers, 2014, 63(2):383-396.
- [14] Tanner H G, Christodoulakis D K. Decentralized cooperative control of heterogeneous vehicle groups[J]. 2007, 55(11):811-823.