A data transmission method for particle physics experiments based on Ethernet physical layer *

HUANG Xi-Ru(黄锡汝)[1,2;1)] CAO Ping(曹平)[1,2;2)] ZHENG Jia-Jun(郑佳俊)[1,2]

[1] State Key Laboratory of Particle Detection and Electronics, University of Science and Technology of China, Hefei 230026, China
[2] Anhui Key Laboratory of Physical Electronics, Department of Modern Physics, University of Science and Technology of China, Hefei 230026, China

Abstract: Due to its advantages of universality, flexibility and high performance, fast Ethernet is widely used in readout system design for modern particle physics experiments. However, Ethernet is usually used together with the TCP/IP protocol stack, which makes it difficult to implement readout systems because designers have to use the operating system to process this protocol. Furthermore, TCP/IP degrades the transmission efficiency and real-time performance. To maximize the performance of Ethernet in physics experiment applications, a data readout method based on the physical layer (PHY) is proposed. In this method, TCP/IP is replaced with a customized and simple protocol, which makes it easier to implement. On each readout module, data from the front-end electronics is first fed into an FPGA for protocol processing and then sent out to a PHY chip controlled by this FPGA for transmission. This kind of data path is fully implemented by hardware. From the side of the data acquisition system (DAQ), however, the absence of a standard protocol causes problems for the network related applications. To solve this problem, in the operating system kernel space, data received by the network interface card is redirected from the traditional flow to a specified memory space by a customized program. This memory space can easily be accessed by applications in user space. For the purpose of verification, a prototype system has been designed and implemented. Preliminary test results show that this method can meet the requirements of data transmission from the readout module to the DAQ with an efficient and simple manner.

Key words: readout method, Ethernet, physical layer, data redirection

PACS: 29.85.Ca **DOI:** 10.1088/1674-1137/39/11/116102

1 Introduction

In modern particle physics experiments, with the increasing demands of large numbers of electronic channels, massive data volume and short readout period, there are great challenges for readout system design [1, 2]. In addition, the newly developed and gradually used waveform digitizing technique also brings enormous pressure for readout system design [3–5]. Readout systems need to concentrate large amounts of data from the front-end electronics (FEE) and send them to the data acquisition system (DAQ) in real-time. Generally, the readout system is comprised of some standard crates (e.g. VME) inside which some readout modules are installed as shown in Fig. 1. Each readout module receives data from the FEE and then sends data to the crate controller one-by-one through the crate backplane bus [2, 6]. Finally, the controller module sends the concentrated data to the DAQ through its Ethernet channel.

This readout scheme is a distributed architecture in which each crate controller acts as a network node and the DAQ acts as a data receiving center. All nodes are connected together by Ethernet. To improve the data transmission performance, Gigabit Ethernet is usually used in this scheme. For the case of experiments with a large number of electronic channels, to balance data transmission performance, designers can adjust the number of readout crates and electronic modules in them.

Unfortunately, there are some weak points in this scheme. Firstly, transmission load balancing puts pressure on readout crates. Readout system designers have to increase either the number of readout crates or the performance of the backplane bus and controller. Secondly, combined with the TCP/IP protocol stack, the

Received 4 March 2015, Revised 14 May 2015

∗ Supported by National Natural Science Foundation of China (11005107) and Independent Projects of State Key Laboratory of Particle Detection and Electronics (201301)

1) E-mail: xiru@mail.ustc.edu.cn

2) E-mail: cping@ustc.edu.cn

efficiency of Ethernet is degraded seriously. To guarantee reliability in TCP/IP, redundant information is encapsulated into each data packet. Obviously, redundant information benefits the transmission reliability but not its efficiency. Actually, in physics experiment applications, because of the simple network connectivity, transmission reliability can easily be guaranteed, but more attention needs to be given to transmission efficiency.

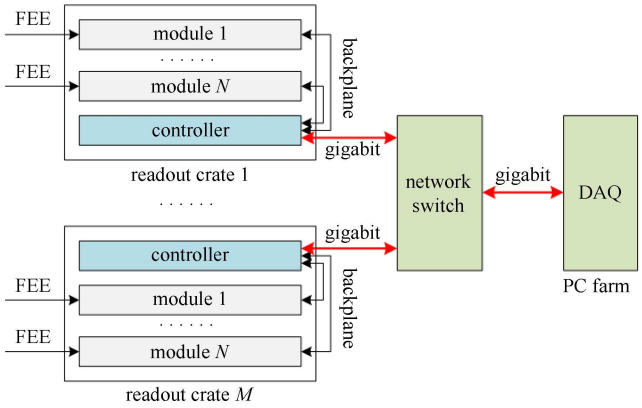

Fig. 1. (color online) Readout scheme based on crates.

Figure 2 shows a different readout scheme. In this scheme, each readout module can send data to the DAQ separately. There are many advantages compared with the traditional scheme. Firstly, there is no direct data transmission path between the readout module and crate controller. The uploaded data from the FEE is transmitted by the readout modules in parallel. This allows crates with a lower performance backplane bus to be used for various experiments at large and small scales. This scheme therefore has an advantage for system expansibility. Secondly, the controller and backplane bus are free from the readout data transmitting flow and switched to the purpose of system control or configuration. In fact, in physics experiment applications, the path of data (received from FEE) uploading differs significantly from that of data (received from DAQ) downloading. The data throughput or real-time performance demands of the former is stricter than the latter. In this new scheme, these two kinds of data path are separated and implemented as a full duplex formation. Therefore, this new scheme does not put pressure on the backplane bus and controller.

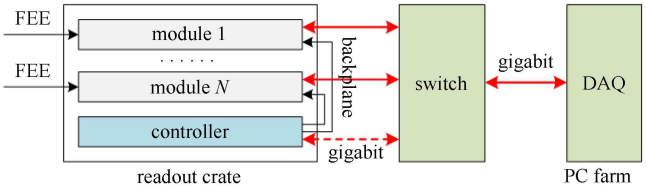

Fig. 2. (color online) Readout scheme based on individual modules.

2 Readout method based on Ethernet PHY

The Ethernet technique plays an important role for realizing this advance readout scheme. As is well known, Ethernet is generally used combined with TCP/IP. To support TCP/IP while implementing an electronic readout module, a CPU is usually used with an embedded operating system [7, 8]. Embedded systems have the disadvantages of increased cost, complexity and power consumption etc. Some experiments use FPGA logic cores to implement this embedded system but it is hard to achieve good stability and performance [9].

Figure 3 shows a new scheme of Ethernet implementation.

In this scheme, data from the FEE is fed into the FPGA on each readout module. According to the processing performance, one readout module can receive data from a number of FEE channels. The FPGA's main task is protocol resolving and data packaging. The repackaged data is then sent to a standalone PHY chip for transmitting over a long cable connected to a DAQ computer. This PHY chip is controlled by the FPGA through a media independent interface (MII). It is a standard interface for Ethernet devices defined by the IEEE-802.3 specification [10]. Based on this readout scheme, data can easily be sent to the DAQ computer through an Ethernet connection.

The FPGA logic architecture is shown in Fig. 4. The data receiving module controls the interface between the readout module and FEE. The raw data from the FEE can be obtained by the protocol resolving module. This protocol is customized between the FEE and readout system for the sake of transmitting data correctly, efficiently and reliably. Depending on the absence or existence of a trigger signal, the data processing module decides whether the received data is valid or not. The data zero compression algorithm can also be implemented in this module. After being processed, valid data is transmitted to the event building module, which will package the valid raw data according to physics experiment requirements. Finally, for the purpose of data transmission over Ethernet PHY, the data should be re-packaged and buffered in local RAM to wait to be read out. The data packet format is shown in Table 1.

In Table 1, Module No. refers to the number of each readout module. It can be configured during system initialization. With the help of this number, the DAQ can construct the relationship between readout modules and FEE channels.

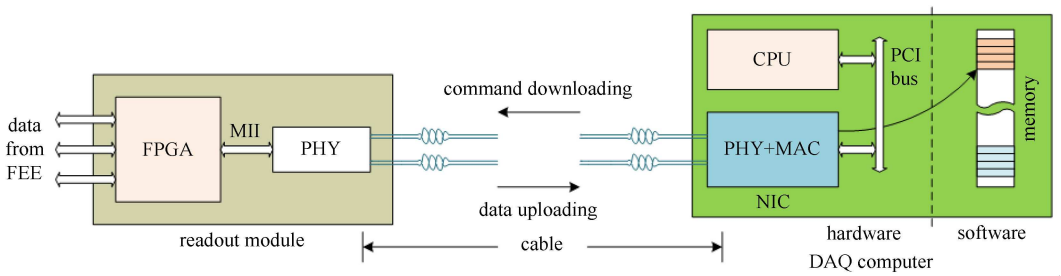

Fig. 3. (color online) Readout scheme based on Ethernet PHY.

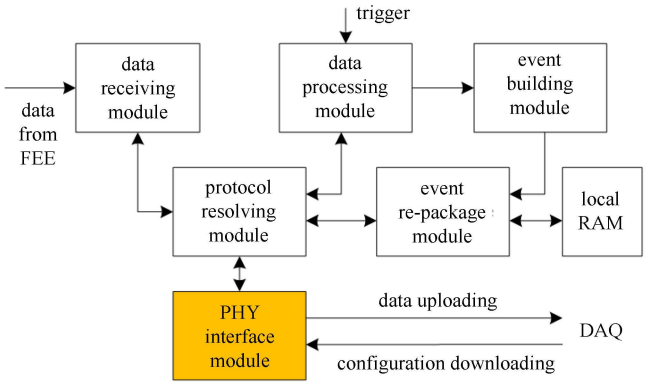

Fig. 4. (color online) FPGA logic Structure.

Counting No. refers to the packet numbers in the data flow. Obviously, the whole data flow is divided into many fragments. One counting number corresponds to one fragment. Once one data fragment packet is missed or does not pass the correctness check, the DAQ can require the readout module to retrieve and re-transmit the corresponding packet according to this number.

Packet size refers to the actual size of raw data in one packet. For the sake of simplicity and efficiency of transmission, data packet size is designed to have a fixed size. It is 1024 in this paper. For the case of one long size data flow (>1024), the last data fragment size may be less than 1024, e.g. 500. The packet size should be 500 and the rest of the data in this packet should be 0.

Raw data refers to the valid data waiting for transmission to the DAQ.

The CRC item refers to the checking data according to the cyclic redundancy check algorithm. If the DAQ finds out that a CRC data mismatch has occurred, it sends a data control command to the readout module to re-transmit the corresponding data fragment.

The PHY interface module is the data swap channel between the readout module and DAQ. It controls the Ethernet PHY to execute the transmitting or receiving function through the MII or GMII interface defined by the IEEE-802.3 protocol.

Figure 5 shows the data transmission state transition chart.

The initial state for data transmission is the idle state. In this state, the transmission state is waiting for valid data to occur. Once valid data is obtained and repackaged, the system enters the transmission state. The duration for the system residing in this state depends on the valid data size. If no error occurs, system enters back into the idle state to finish the transmission procedure. If any error (checking data mismatch) occurs, however, the system enters into the re-transmission state. The corresponding data fragments will be re-transmitted to the DAQ.

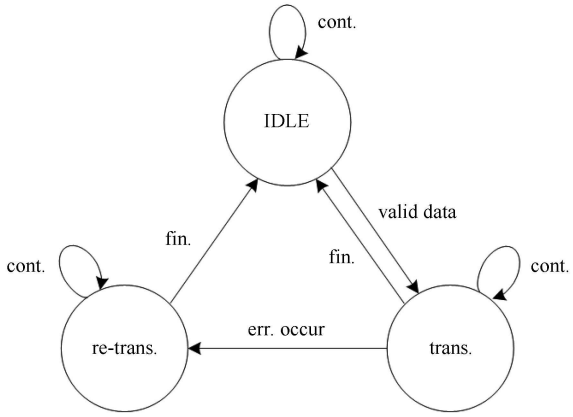

Fig. 5. Data transmission state transition chart.

From the readout module side, the real-time performance and transmission efficiency can easily be guaranteed because the whole data path is implemented by full hardware without TCP/IP and because the transmitted data over Ethernet is nearly all raw data from the FEE with only a little redundant information. However, from the DAQ side, the data packets from the readout module cannot be recognized correctly because of the absence of standard TCP/IP. To solve this problem, the standard data flow of an ordinary network interface card (NIC) should be modified.

In Linux operating system kernel space, the network device driver receives the data from PHY and sends it to the TCP/IP stack processing module. Generally, only after passing through all the protocol layers one-by-one

can data be resolved correctly and eventually sent to applications in user space, as shown on the left side of Fig. 6.

To simplify the whole system design as much as possible, the easiest way to make the DAQ applications in user space recognize the received data flow correctly is to redirect the data flow, bypassing the TCP/IP processing module. As shown on the right side of Fig. 6, in the operating system kernel space, the data is redirected to a new path. The raw data processing module will decapsulate and process the data according to a customized protocol and then buffer the data into a memory which is mapped to user space during the initialization state. Once data is de-capsulated and stored in these buffers, user applications can access it directly. Now, from the point of view of the device driver, the network interface card acts as a normal character device, not a network device. The driver is simplified.

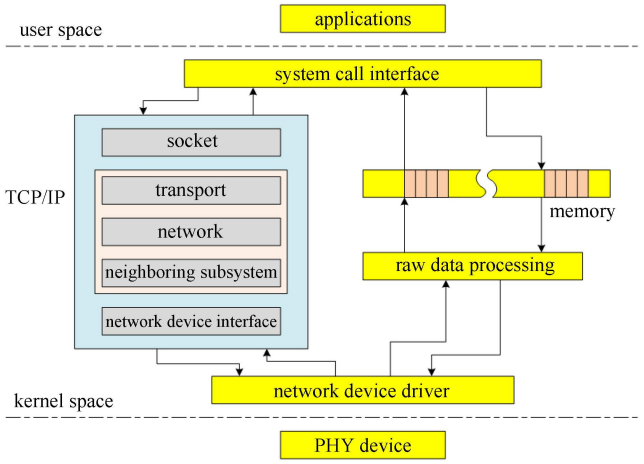

Fig. 6. (color online) Data flow redirection.

Although the DAQ can communicate with the readout module with this modified device driver, when it communicates with other computers, there will be the problem of device driver compatibility. So the driver should also be able to recognize whether the data packet is coming from the customized readout module or from normal computers.

Actually, Ethernet was designed before the IEEE created its 802.3 standards. The latter is not pure Ethernet, even though it is commonly called Ethernet standards. Fortunately, every Ethernet card is able to receive both the 802 standard frames and the old Ethernet frames. The main difference between these two kinds of network standard is the header of the Ethernet frame format as shown in Fig. 7. The number above each item box refers to the size of this item in the frame [11].

To save space, the IEEE decided to use values greater than 1536 to represent the Ethernet protocol. The 802.3 protocol, however, uses the field to store the length of the frame. This item is 2 bytes in size. The valid Ethernet types are predefined by IEEE and cannot be used arbitrarily. So, in this paper, we define the protocol type to be 0xFF00 which is different from the existing types. The data domain refers to the format defined in Table 1. The size is fixed to be 1024. Once the network device driver receives packets with protocol type 0xFF00, it redirects them to the new customized raw data processing module in the kernel space. In this way, the TCP/IP stack is bypassed.

Table 1. Raw data packet format.

item	size (byte)
module No.	1
counting No.	1
packet size	1
raw data	···
CRC	1

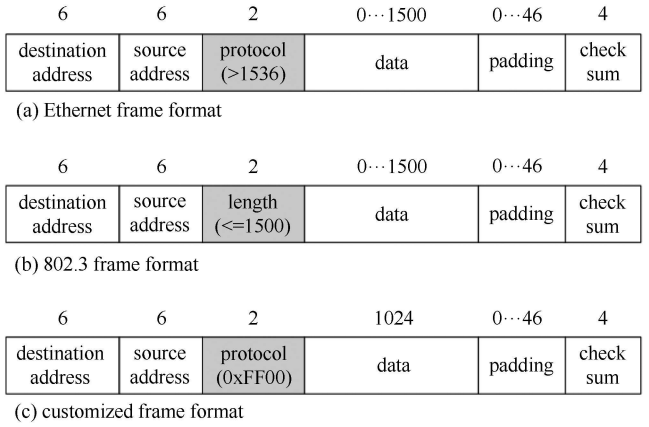

Fig. 7. Differences between Ethernet, 802.3 and customized frames.

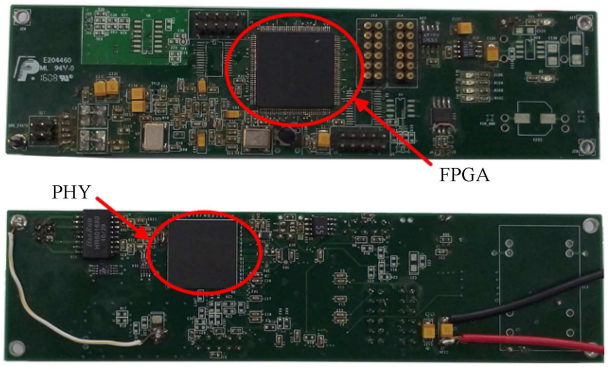

Fig. 8. (color online) Prototype readout board.

3 Verifications

To verify this proposed readout method, a prototype readout board has been implemented as shown in Fig. 8.

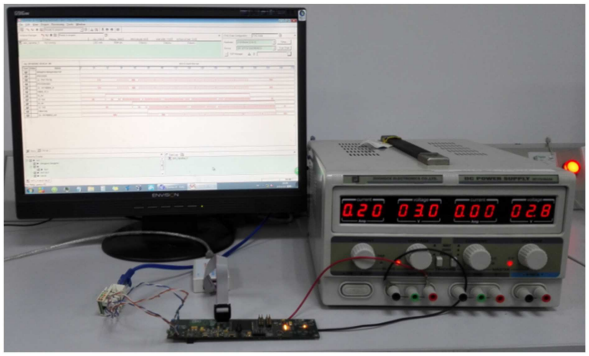

Fig. 9. (color online) Test platform.

During verification, the FPGA (a low cost Altera Cyclone chip) generates data and sends it out through the LVDS port to simulate the behavior of the FEE. Then the data is looped through the LVDS input port back to the FPGA to simulate the behavior of the readout module receiving data. The data is re-packaged in the FPGA and then sent to an Ethernet PHY chip (ANX5802, 100Mbps). Finally, the PHY chip sends data to the DAQ computer.

The test platform is shown in Fig. 9. Test results show that this prototype of readout module can receive data from the FEE and send to the DAQ computer correctly and smoothly. The data throughput can reach up to 100 Mbps.

4 Conclusions

In this paper, a novel and simple data readout method is proposed for the applications of physics experiments. The key point of this method is to try to obtain good simplicity and performance for Ethernet without TCP/IP. The absence of the TCP/IP stack allows the readout module to be easily implemented, i.e. an FPGA combined with a PHY chip is enough for implementation. A simple and effective data re-transmission scheme can guarantee the reliability of data transmission. To make the non-standardized data packets be recognized correctly by the DAQ software, a data flow redirection method is realized in operating system kernel space. To verify this method, a prototype readout board is implemented. Test results have verified the feasibility and correctness of this method.

Besides physics experiment applications, this method can also be used in other areas which require data transmission over Ethernet.

References

1 Essel H G. IEEE Trans. Nucl. Sci., 2006, **53**(3): 677–681
2 BESIII Design Report. http://bes.ihep.ac.cn/bes3/design05/design/design1.htm
3 Abbasi R, Ackermann M, Adams J et al. Nuclear Instruments and Methods in Physics Research A, 2009, **601**(3): 294–316
4 Akerib D S, BAI X, Bedikian S et al. Nuclear Instruments and Methods in Physics Research A, 2012, **668**: 1–8
5 Agnes P, Alexander T, Alton A et al. arXiv: astroph.IM/1412.2969
6 Harris F A, Kennedy J W, LIU Q et al. Nuclear Instruments and Methods in Physics Research A, 2008, **593**(3): 255–262
7 Axelson J. Embedded Ethernet and Internet Complete. Madison, Wisconsin: Lakeview Research, Inc., 2003. chapter 2–5
8 Scaglia S. The Embedded Internet: TCP/IP Basics, Implementation and Applications. Boston, Massachusetts: Addison-Wesley Professional, 2007. chapter 12–13
9 LIU M, LANG J, YANG S et al. ATCA-based Computation Platform for Data Acquisition and Triggering in Particle Physics Experiments. In: Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on. IEEE, 2008: 287–292
10 IEEE 802.3™-2012. http://standards.ieee.org/about/get/802/802.3.html
11 Benvenuti C. Understanding Linux Network Internals. First edition. Sebastopol, California: O′Reilly Media, Inc., 2006. 283