

Research on A Reliable Method of Receiving Data Based on Zero-Copy Non-Drop Packet Network Card

Yujian Gong¹, Wei Xi², Hui Han¹, Jiawei Yan¹

¹ Beijing Acoinfo Technology CO., LTD, Beijing, China

² Electric Power Research Institute, CSG, Guangzhou, China

gongyujian@acoinfo.com, xiwei@csg.cn, acoinfo@acoinfo.com, yanjiawei@acoinfo.com

Abstract— Under the TCP protocol, lost data packets will be retransmitted, but under the UDP protocol, packets will be discarded, resulting in a waste of network bandwidth and an increasing of the instability of the network to a certain extent. This paper proposes a method of receiving data based on receiving zero-copy non-drop packet network card. When the zero-copy pool resources are exhausted during the network card receiving process, a static pool is used to ensure normal data reception and avoid packet dropping problems.

Keywords— Receiving Zero-Copy, Non-Drop Packet, Network Card Receiving Data, Descriptor

I. INTRODUCTION

For receiving zero copy, the general processing method is to reserve a memory area when the system starts, that is, the zero-copy pool, and the CPU allocates memory resources from the zero-copy pool. There may be an allocation failure, which causes the network card descriptor unable to obtain available memory resources, and the data cannot be received, resulting in the data packets being dropped out^{[1][2]}. Under the TCP protocol, lost data packets will be retransmitted, but under the UDP protocol, packets will be dropped out. As a result, the network bandwidth will be wasted to a certain extent and the network instability will be increased^{[3][4]}.

This paper proposes a method of receiving data based on receiving zero-copy non-drop packet network card, which includes: starting the system, creating a zero-copy pool, configuring receiving descriptors, and allocating N memory resources according to the number of configured receiving descriptors N. It is managed uniformly and a static pool is formed. The memory resources in the static pool is mapped to the receiving descriptors one by one. Then, determine whether the resources are obtained from the zero-copy pool or the static pool. If it is the zero-copy pool, the pointing memory resource from the current receiving descriptor is directly submitted to the protocol stack. Otherwise, if it is from the static pool, the pointing resource obtained from the static pool from the current receiving descriptor is copied to the protocol stack. In the data receiving process of the network card, when the resources of the zero-copy pool are exhausted, a static pool

is used to ensure normal data reception and avoid packet drop problems.

II. STATUS AND ANALYSIS OF NETWORK CARD DATA RECEPTION

There are many optimizing methods for various levels in order to improve the overall network throughput. The optimization of the link layer mainly lies on the network card driving. The main method of network card driving optimization is to reduce the number of memory copies during data transmission and reception, that is, zero-copy technology, which includes the protocol stack delivers the data to the network card and drives the transmitting process, in which the zero-copy is sent. The network card driver receives the data and submits to the protocol stack, and the zero-copy is received.

However, for receiving zero-copy, the general processing method is to reserve a memory area when the system starts, that is, the zero-copy pool. The CPU allocates memory resources from the zero-copy pool, saves the virtual address of the memory resources, and returns the physical address of memory resources to the network card receiving descriptor. When the network card receives the data, it will place the data in the corresponding memory resources of receiving descriptor, and notifies the CPU. Then the CPU directly submits the memory resources to the network protocol stack through the previously saved virtual address, therefore achieving zero-copy reception.

Nonetheless, there is a possibility of allocation failure for the CPU to allocate memory resources from the zero-copy pool, which will make the network receiving descriptor unable to obtain available memory resources, cannot receive data and the data packets will be discarded. Under the TCP protocol, lost data packets will be retransmitted, but under the UDP protocol, packets will be discarded, resulting in a waste of network bandwidth and an increasing of the instability of the network to a certain extent.

III. FUNDAMENTAL PRINCIPLES

This paper proposes the method of receiving data based on receiving zero-copy non-drop packet network card, and the process is as follows:

Step 1: Start the system, create a zero-copy pool, configure the receiving descriptors, and allocate N memory resources according to the configured receiving descriptor number N , and manage them uniformly to form a static pool, where N is a natural number and $N \geq 1$.

Step 2: Map the memory resources in the static pool to the receiving descriptors one by one. Each receiving descriptor has a unique memory area in the static pool.

Step 3: In the process of receiving the data of the network card, enter the receiving service program from the interrupt to determine whether the resource is obtained from the zero-copy pool or the static pool. If it is the zero-copy pool, the pointing memory resource from the current receiving descriptor is directly submitted to the protocol stack. Otherwise, if it is from the static pool, the pointing resource obtained from the static pool from the current receiving descriptor is copied to the protocol stack.

Further, configuring the receiving descriptor in step 1 includes: when initializing the receiving descriptor, preferentially obtaining the memory resource from the zero-copy pool, and marking that the descriptor m uses a zero-copy pool resource, where $1 \leq m \leq N$. If obtaining the memory resource according to the receiving descriptor n is unsuccessful, obtain the corresponding memory resource from the static pool, and mark the descriptor n as using the static pool resource, where $1 \leq n \leq N$.

Further, the method for determining whether the resources are obtained from the zero-copy pool or the static pool in step 3 includes: determining whether to obtain resources from the zero-copy pool or the static pool according to the resource type of the current receiving descriptor. If the resource is obtained from the zero-copy pool, the pointing memory resource from the current receiving descriptor is directly submitted to the protocol stack. Otherwise, if it is obtained from the static pool, the pointing resource from the static pool of the current receiving descriptor is copied to the protocol stack. Then the receiving descriptor is reset and the data receiving process ends.

Further, step three further includes: in the process of receiving the data of the network card, entering the receiving service program from the interrupt, first obtaining resources from the zero-copy pool. If the resources can be obtained from the zero-copy pool, the pointing memory resource from the current receiving descriptor m is directly submitted to the protocol stack, and the descriptor m uses zero-copy pool resources, where $1 \leq m \leq N$. Otherwise, if it can be obtained from the static pool, the pointing resource from the static pool of the current receiving descriptor n is copied to the protocol stack, and mark that the receiving descriptor n uses a static pool resource, where $1 \leq n \leq N$.

Furthermore, the process after copying the pointing resource from the static pool of the current receiving descriptor to the protocol stack in step 3 includes: before

exiting the receiving service program, resetting the receiving descriptor so that the receiving descriptor is available.

Further, resetting the receiving descriptor includes: during the network card data receiving process, entering the receiving service program from the interrupt, first obtaining resources from the zero-copy pool. If resources are successfully obtained from the zero-copy pool, initialize the receiving descriptor and mark that zero-copy pool resources are used. Otherwise, if the resources are obtained from the static pool, initialize the receiving descriptor, mark that the static pool resources are used, reset the receiving descriptor and the process ends.

It should be noted that this article involves computer readable storage medium, in which computer program instructions are stored. When the computer program instructions are executed by a processor, the corresponding methods are implemented.

IV. CONCRETE IMPLEMENTATIONS

A. General Framework

The method of receiving data based on receiving zero-copy non-drop packet network card in this paper mainly includes three aspects, creating a static pool, receiving data to submit the protocol stack, and resetting the receiving descriptor after receiving data.

B. Creating A Static Pool

The method for creating a static pool is shown in Figure 1. The method includes the following steps:

Step 1: Start the system, create a zero-copy pool, configure the receiving descriptor, and allocate N memory resources according to the configured receiving descriptor number N , and manage them uniformly to form a static pool, where N is a natural number and $N \geq 1$.

Step 2: The memory resources in the static pool are mapped one by one with the receiving descriptors. Each receiving descriptor has a unique memory area in the static pool.

Configuring the receiving descriptor includes: when initializing the receiving descriptor, preferentially obtain memory resources from the zero-copy pool, and mark the descriptor m as using the zero-copy pool resource, where $1 \leq m \leq N$. If obtaining the memory resource according to the receiving descriptor n is unsuccessful, obtain the corresponding memory resource from the static pool, and mark the descriptor n as using the static pool resource, where $1 \leq n \leq N$.

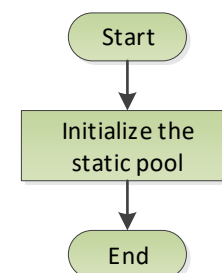


Figure 1. Procedure of creating a static pool

C. Resetting the Receiving Descriptor

In the process of receiving data from the network card, enter the receiving service program from the interrupt, first obtain resources from the zero-copy pool. If resources are successfully obtained from the zero-copy pool, initialize the receiving descriptor and mark that zero-copy pool resources are used. Otherwise, if the resources are obtained from the static pool, initialize the receiving descriptor, mark that the static pool resources are used, reset the receiving descriptor and the process ends.

Before exiting the receiving service program, reset the receiving descriptor to make the receiving descriptor available. The concrete method of resetting the receiving descriptor is shown in Figure 2.

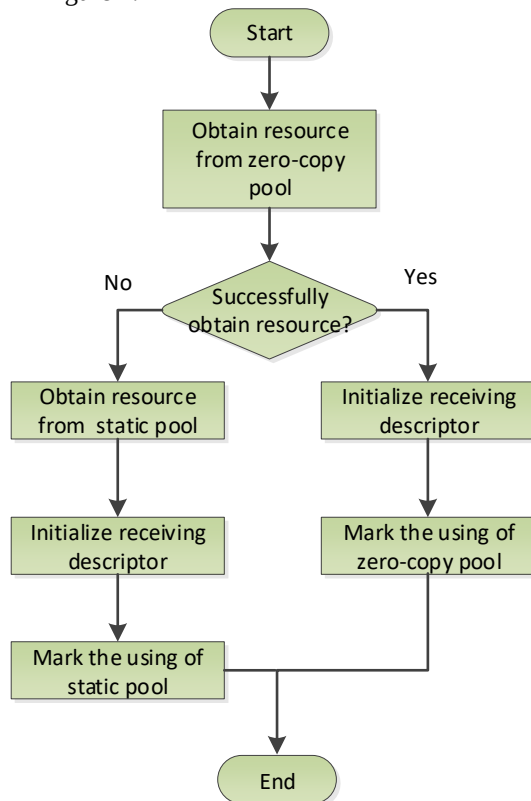


Figure 2. Procedure of resetting the receiving descriptor

D. Data Receiving of Non-Drop Packet Network Card

Figure 3 is a flowchart of receiving data based on receiving zero-copy non-drop packet network card in this paper. The method includes the following steps:

In the process of receiving network card data, it can be determined whether the resource is obtained from the zero-copy pool or the static pool according to the resource type of the current receiving descriptor. If the resource is obtained from the zero-copy pool, the pointing memory resource from the current receiving descriptor is directly submitted to the protocol stack. Otherwise, if it is obtained from the static pool, the pointing resource from the static pool of the current receiving descriptor is copied to the protocol stack. Then the receiving descriptor is reset and the data receiving process ends.

Resetting the receiving descriptor includes: first obtaining resources from the zero-copy pool. If resources are successfully obtained from the zero-copy pool, initialize the receiving descriptor and mark that zero-copy pool resources are used. Otherwise, if the resources are obtained from the static pool, initialize the receiving descriptor, mark that the static pool resources are used, reset the receiving descriptor and the process ends.

In addition, in the process of receiving data from the network card, enter the receiving service program from the interrupt, first obtain resources from the zero-copy pool. If the resources can be obtained from the zero-copy pool, the pointing memory resource from the current receiving descriptor m is directly submitted to the protocol stack, and the descriptor m uses zero-copy pool resources, where $1 \leq m \leq N$. Otherwise, if it can be obtained from the static pool, the pointing resource from the static pool of the current receiving descriptor n is copied to the protocol stack, and mark that the receiving descriptor n uses a static pool resource, where $1 \leq n \leq N$.

If the memory resources using the static pool cannot be directly submitted to the protocol stack, that is, when the CPU fails to allocate memory resources from the zero-copy pool, the corresponding memory resources of the receiving descriptor can be obtained from the static pool, and it must be successfully obtained to ensure that the receiving descriptor can receive data normally, and then submitted to the network protocol stack through one copy.

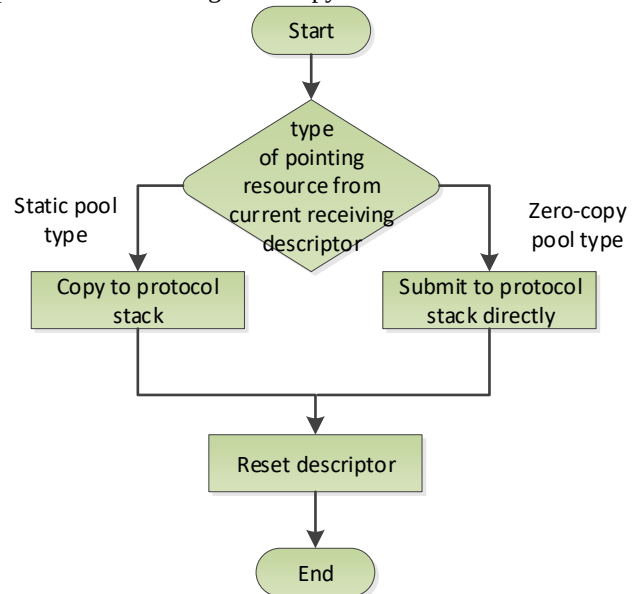


Figure 3. Procedure of receiving data based on receiving zero-copy non-drop packet network card

E. Experimental Verification

Experimental environment:

The test device runs a test program. The function of the test program is to receive UDP data packets from the PC test machine and transmit them to the PC test machine. The PC test machine uses the NetAssist tool to send data and receive

the data transmitted by the test device, and then observe sending and receiving data of PC test machine.



PC test machine

Figure 4. Diagram of PC test machine and test device

The testing program of the device is as follows:

```

#include <stdio.h>
#include <string.h>
#include <sys/socket.h>

/*****
*****
***** 宏定义
*****
*****
#define __UDP_ECHO_PORT_SERVER      8001
/* 服务器端口号 */
#define __UDP_ECHO_BUFF_SIZE_SERVER 1500
/* 服务器接收缓冲区大小*/
/*****
***** 函数名称: __UdpEchoServer
** 功能描述: UDP 服务器
** 输入 : NONE
** 输出 : 错误号
** 全局变量:
** 调用模块:
*****
*****/
static int __UdpEchoServer (void)
{
    int iRet = -1;
    int sockFd = -1;
    socklen_t uiAddrLen = sizeof(struct sockaddr_in);
    register ssize_t sstRecv = 0;
    char
cRecvBuff[__UDP_ECHO_BUFF_SIZE_SERVER] = {0};
    struct sockaddr_in sockaddrinLocal;
    struct sockaddr_in sockaddrinRemote;

    fprintf(stdout, "UDP echo server start.\n");

    sockFd = socket(AF_INET, SOCK_DGRAM,
IPPROTO_UDP);
    if (sockFd < 0) {
        printf("UDP echo server socket error.\n");
        return (-1);
    }

    /*
    * 初始化本地地址结构
    */
    memset(&sockaddrinLocal, 0, sizeof(sockaddrinLocal));

    sockaddrinLocal.sin_len = sizeof(struct
sockaddr_in);
    sockaddrinLocal.sin_family = AF_INET;
    sockaddrinLocal.sin_addr.s_addr = INADDR_ANY;
    sockaddrinLocal.sin_port =
htons(__UDP_ECHO_PORT_SERVER);

    iRet = bind(sockFd, (struct sockaddr
*)&sockaddrinLocal, sizeof(sockaddrinLocal));
    if (iRet < 0) {
        close(sockFd);
        fprintf(stderr, "UDP echo server bind error.\n");
        return (-1);
    }

    for (;;) {
        memset(&cRecvBuff[0], 0,
__UDP_ECHO_BUFF_SIZE_SERVER);
        sstRecv = recvfrom(sockFd, (void *)&cRecvBuff[0],
__UDP_ECHO_BUFF_SIZE_SERVER,
0,
(struct sockaddr *)&sockaddrinRemote,
&uiAddrLen);
        if (sstRecv <= 0) {
            if ((errno != ETIMEDOUT ) &&
(errno != EWOULDBLOCK)) {
                close(sockFd);
                fprintf(stderr, "UDP echo server recvfrom
error.\n");
                return (-1);
            }
            continue;
        }

        sendto(sockFd,
(const void *)&cRecvBuff[0],
sstRecv,
0,
(const struct sockaddr *)&sockaddrinRemote,
uiAddrLen);
    }

    return (0);
}

/*****
*****
***** 函数名称: main
** 功能描述: main 函数
** 输入 : argc, argv[]
** 输出 : 错误号
** 全局变量:
** 调用模块:
*****
*****/
  
```

```

int main (int argc, char *argv[])
{
    __UdpEchoServer();

    return (0);
}

```

Testing method 1:

The device under test does not run this method. When the zero-copy pool resource is exhausted during the network card receiving data, the static pool is not used.

Test results:

When the zero-copy pool resources of the test device are exhausted, the data is actively discarded, resulting in a reduction in the amount of data forwarded by the test device, which makes the amount of data received by the PC test machine much less than the amount of sent data.

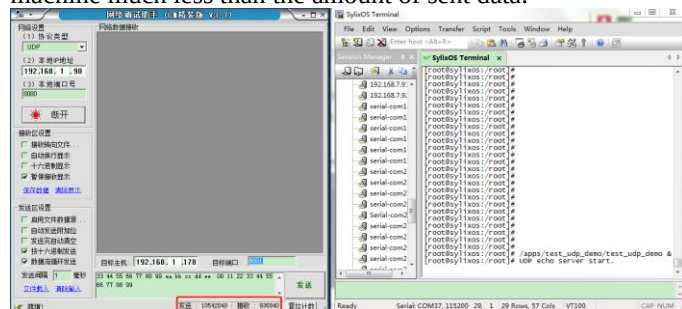


Figure 5. Interface of test method 1

Test method 2:

The test device runs with this method, and uses the static pool when the zero-copy pool resources are exhausted when the network card is receiving data.

Test results:

When the zero-copy pool resources of the test device are exhausted, static pool resources are used, and the test device normally forwards data, so that the amount of data received by the PC test machine is equivalent to the amount of sent data. The amount of received data is not equal to the amount of sent data, because the UDP protocol is unreliable, and this method avoids the link layer packet from being discarded.

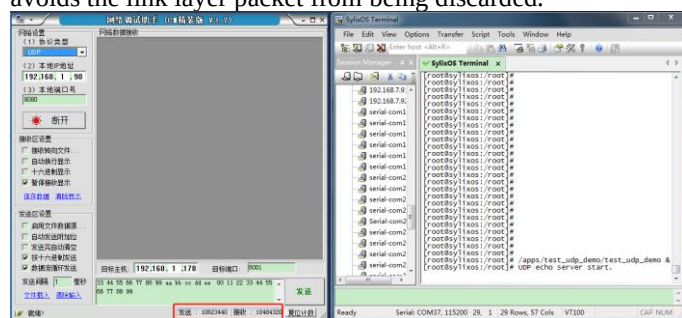


Figure 6. Interface of test method 2

F. Results

In the technical solution in this paper, the concept of static pool is proposed during the process of receiving data by the network card, which complements the zero-copy pool. When

the resources of the zero-copy pool are exhausted, normal data reception will be guaranteed and packet dropping problems will be avoided.

V. DISCUSSION

This paper analyzes the general processing methods of receiving zero-copies from the existing technology, and draws the problems in the current study: the CPU allocates memory resources from the zero-copy pool, and there is a possibility of allocation failure, which will make the network card receiving descriptor unable to obtain available memory resources, data cannot be received and data packets are forced to be dropped. Under the TCP protocol, lost data packets will be retransmitted, but under the UDP protocol, packets will be discarded, resulting in a waste of network bandwidth and an increasing of the instability of the network to a certain extent.

Furthermore, a method of receiving data based on receiving zero-copy non-drop packet network card is proposed, which includes: starting the system, creating a zero-copy pool, configuring receiving descriptors, and allocating N memory resources according to the number of configured receiving descriptors N . It is managed uniformly and a static pool is formed. The memory resources in the static pool is mapped to the receiving descriptors one by one. Then, determine whether the resources are obtained from the zero-copy pool or the static pool. If it is the zero-copy pool, the pointing memory resource from the current receiving descriptor is directly submitted to the protocol stack. Otherwise, if it is from the static pool, the pointing resource obtained from the static pool from the current receiving descriptor is copied to the protocol stack. In the data receiving process of the network card, when the resources of the zero-copy pool are exhausted, a static pool is used to ensure normal data reception and avoid packet drop problems.

Based on the understanding, the technical solution in this paper is essentially a part that contributes to the existing technology, or all or part of the technical solution can be embodied in the form of a software product, which is stored in a storage medium, including several instructions to make a computer device (which can be a personal computer, a server, or a network device, etc.) perform all or part of the steps in the technical solution in this paper. The above-mentioned storage media includes: U disks, mobile hard disks, read-only memory (ROM), random access memory (RAM), magnetic disks, or optical disks and other media that can store program codes.

ACKNOWLEDGMENT

This work is supported by National Key R&D Program of China(2018YFB0904900, 2018YFB0904902).

REFERENCES

- [1] Du Haipeng, Zheng Qinghua, Zhang Weizhan, etc. A Packet Loss Differentiation Algorithm for 4G-LTE Network[J]. Journal of Computer Research & Development, 2015.
- [2] Xu Kailu. Design and Implementation of the Network Packet Loss Rate Measurement System [D]. Xidian University, 2014.
- [3] Liu Baochen. Research and Implementation of a High-Performance Packet Capture System [D]. Shanghai Jiao Tong University, 2013.

- [4] Shengli L . Zero-copy communication protocol based on the real-time micro-kernel[J]. Journal of Huazhong University of Science & Technology, 2007, 35(1):35-38.
- [5] Xiao-Dong S , Qiang L . A strategy for continuously big capacity data transmit/receive process handling in real-time[C]// International Conference on Advanced Computer Control. IEEE, 2010.
- [6] Chao Wang. The design and implementation of the general drive system for the zero copy technology of Ethernet card[D]. University of Chinese Academy of Sciences(School of Engineering Science), 2015.



Yujian Gong was born in Chifeng, Inner Mongolia, China in 1985. He is currently an engineer in Beijing Acoinfo Technology CO., LTD. His research interests include key technologies of embedded real-time operating system, and he focuses on the research and engineering application of high stability, reliability and security of power industry and power system. Email: gongyujian@acoinfo.com.

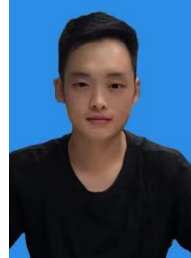


Wei Xi was born in Jianli, Hubei province, China in 1981. He graduated from Huazhong University of Science and Technology and is a professor-level senior engineer now in Digital Grid Research Institute, CSG., Guangzhou, China. His research interests include power system relay

protection, digital transformer substation, smart grid and electric power chip development. Email: xiwei@csg.cn.



Hui Han was born in Xi'an, Shaanxi province, China in 1983. He is currently an engineer in Beijing Acoinfo Technology CO., LTD. His research interests include embedded real-time operating system inter-core research and general architecture design, and mainly focuses on RTOS core technology with high reliability and high real-time requirements. Email: acoinfo@acoinfo.com.



Jiawei Yan was born in Wuzhong, Ningxia province, China in 1993. His research interests include embedded real-time operating system and application of operating system in power field. Email: yanjiawei@acoinfo.com.