

Research on software developing model of NetMagic Platform

Sun Kai¹, Xu Donglai², Mao Jianbiao³, Sun Zhigang⁴, SuXing⁵

¹⁻⁴School of Computer, National University of Defense Technology, Hunan 410073, Changsha, China

⁵College of Information Technology, Jiangsu 223003, Huaian, China

365604512@QQ.com, xudonglai0923@163.com, mjn198812@sina.com, sunzhigang@263.net, sxbox@163.com

Abstract - As a programmable network platform, NetMagic have being widely used in network research. But how to develop software on NetMagic is not well studied. A 3- layer model, which include application layer, application supporting layer and NetMagic access layer, is proposed in this paper. Using quality monitoring of streaming media transportation on NetMagic platform as an example, we analyze how to use the 3- layer model in NetMagic software developing.

Index Terms - NetMagic ; Software ; developing model

1. Background

NetMagic is a network-experiment oriented innovative programmable platform [1], Researchers can commit control behavior through inner-FPGA programming to fulfil various functions. Different from the network interface card-alike NetFPGA platform, NetMagic is an independent body which contains eight Ethernet interface, and exists in the form of Ethernet switch. For the reason that simple hardware-based programming and flexible control can be done in NetMagic UM (User Module), the platform can be a good support for network technology research, as a result it has been applied in the summit conference paper writing such as SIGCOMM Demo and CoNEXT [3][4][5].

The software of NetMagic is running on the device-independent computer (called NetMagic control node) which communicates with the NetMagic hardware through NMAC (NetMagic Access Control) protocol [6]. Similar to the master software running on the host CPU of routers, NetMagic software mainly achieve the following three types of functions. (1) handling internal rules and configuring & monitoring NetMagic hardware status; (2) functions not feasible to implement in hardware data plane, such as data compression and decompression etc. (3) the control plane functions in experiments, such as communication with the remote control node.

NMAC is NetMagic control protocol based on Libnet / libpcap [7], all of the software accessible resources in the the NetMagic UM hardware (such as RAM, registers, counters, etc.) are mapped to the unified NetMagic address space. The software on the control node completes internal UM resource management and configure through Read / Write function provided by NMAC. However, as NMAC merely provides configuration & management channels for the NetMagic software development, how to satisfy demand for different

network experiments, and implement fast NetMagic software deployment have not get researched.

This paper proposed a NetMagic software development method, the main contributions include: (1) Proposed the three-layer model for NetMagic software developing which divide software into the application layer, the application of the support layer and NetMagic access control layer from top to bottom. The hierarchical model provides multiple levels of abstraction to simplify the complexity of development; (2) Regarding Application support layer as the core of software development, making application support layer API-based application design more portable; (3) Taking the streaming transmission quality monitoring applications called StreamMon for example to make a detailed description of the design method which centres the application support software. The above work is of important guiding significance to the network experimental design and related software development based on NetMagic platform

2. The Research Goal

The NetMagic software development approach should have the following characteristics. (1) Utilizing a layered architecture to reduced design complexity and improve code reuse rate through multiple levels of abstraction; (2) Supporting the experimental program portability, which means experiment code on NetMagic platform can be easily integrated into other experimental platform; (3) Supporting multiple operating systems, that is, the NetMagic control software can be developed either in Linux-based platform, or Windows and other platforms.

3. The Three-layer Development Model

In Order to achieve the objectives of the above, we propose the layered development model which contains the application layer, application support layer and access control layer as shown in Figure 1. All software developments are based on Libnet / libpcap and socket interface, which is not associated with the operating system.

Access control layer leverages the NMAC protocol to access NetMagic internal RAM table, registers and counters. The typical RAM table contains rule table (such as the flow table in openflow) which control packet handling behavior, a router forwarding FIB (Forward Information Base) table etc.

Typical registers consist of the interface control registers, the UM configuration registers. The typical counters contain interface transmission and reception counter etc.

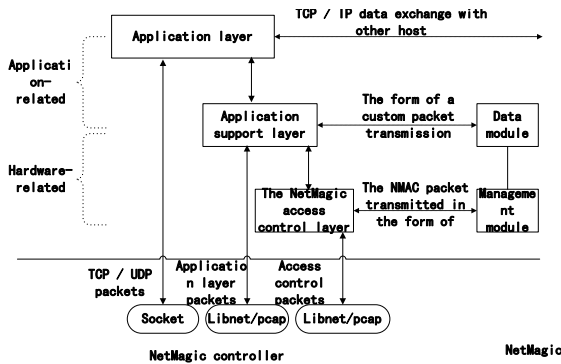


Figure 1 Implementation Routing of the Layered Software

The application support layer implements significant software program which provides NetMagic’s support for experimental applications. On the one hand, by the NMAC protocol it can communicate with NetMagic platform and make configuration, management and maintenance to rules of NetMagic platform in accordance with requirements of the application layer. On the other hand, by Libnet / Lipcap NetMagic it can communicate directly with NetMagic to achieve packet channels between control software and NetMagic hardware implementation.

The application layer is the software program to implement network experiment and has no relationship with NetMagic platform. Software in application layer can interact with NetMagic platform through the interfaces provided by application support layer, at the meanwhile application layer can communicate with other hosts on the network via a standard socket mechanism.

The NMAC layer in the software development model corresponds with the management module of NetMagic hardware to realize the control flow information interaction between hardware and software. According to the NetMagic UM-based hardware development method, UM module is experiment related, that is, different experiment has different UM design. As a result the protocol and message format adopted by communication between application support layer and UM is also experiments related, which means users could define the experimental requirements based on their own experimental needs. Application layer software is experiment related and platform independent, it can communicate with the remote host via a standard TCP / IP protocol.

4. Application Support Layer Design

In the NetMagic software hierarchy, since has relationships with platforms and applications at the same time, the application support layer is actually the core of the design, mainly in:

(1) Provide interfaces to application development and shield the implementation details of NetMagic platform, so

applications based on the API can be easily ported to other experimental platform (provided that other platforms provide the same API), if NetFPGA, NetMagic and commodity platform all provide self-defined API, then StreamMon program can migrated among these platforms without any modification.

(2) Application support layer try its best to simplify the design of the application, Such as the StreamMon software described in Section 5, Support layer needs to compute MDI according to the application demand, make formation of the summary information including matched flow ID, receiving timestamp and packet length for each received packet, the summary information instead of the received packet is passed directly to the application.

(3) Application support layer is related to UM achievement of NetMagic hardware, I.e. In experiments the division of functions is done between UM hardware and software support layer, for example timestamp of receiving packets can be recorded either in hardware UM records, or by application support layer which calls the system function to achieve, and how to make the division between hardware and software closely relates to the complexity of implementation and performance.

Therefore, the application support layer is the core of the NetMagic software design. Software development centring Application support layer includes the following steps:

(1) Determine the application support API; (2) Determine functions implemented in the UM hardware and software support layer; (3) Determine the UM communication protocol and message format; (4) Determine the internal key data structure, functional modules and data flow. These steps will be illustrated in Section 5

5. Example of software development—StreamMon

A. Basic Functions of StreamMon

StreamMon is a streaming media transmission quality monitoring application developed on the NetMagic platform, the basic operating environment is shown in Figure 2. Where A is NetMagic controller which runs StreamMon application, remote server requires StreamMon monitor to calculate the MDI value of the video stream sent from B to C, and return the value in real-time.

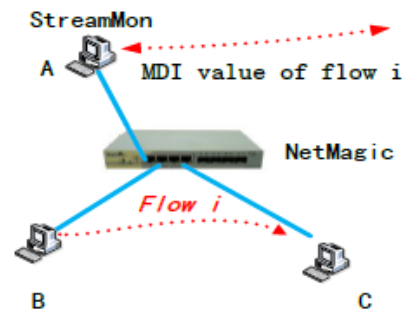


Figure 2 StreamMon Scenes Work

B. StreamMon Application Support Layer Design

Step 1: Design the API Interface of the Application Support Layer

Application needs to select the monitoring data stream, it is therefore necessary to provide the appropriate API of rules management interface, at the meanwhile the application which makes MDI calculations requires to obtain the go through time, serial number and packet length and other information for the specified data stream packets received by NetMagic, as a result the application support layer need to have an API interface to provide the information. The designed application support layer API is as shown in Table 1.

TABLE 1 Support layer on the top of the API

Function name	Meaning
<i>Open_rule_table(rule)</i>	<i>Start the entry function</i>
<i>Add_rule_table(rule)</i>	<i>Add rule function</i>
<i>Del_rule_table(rule)</i>	<i>Delete rule function</i>
<i>Modify_rule_table(rule)</i>	<i>Modify the rule function</i>
<i>Usercallback(pkt_info)</i>	<i>Application callback function</i>

Step 2: Hardware and Software Interface Boundary Division

In this application, hardware and software boundary division involves two functions. First, the packet-flow filtering, that is, only the packets of the data stream concerned by applications is sent to the application layer. Secondly, time stamp. In order to reduce the software processing overhead, rules table are set in hardware, only the packets which meet the rules is sent to NetMagic controller, and that means packet filtering is implemented in hardware. At the meanwhile, because MDI calculation only requires us level accuracy for packet timestamp, software support layer can implement that instead of hardware with ns level accuracy.

Step 3: Determine the UM Communication Protocol and Message Format

For the reason that Timestamp information is marked by the software, NetMagic do not need to modify the captured packets therefore communication message format is set to the non-modified Ethernet packet format. Because the message is not NMAC protocol packets, application support layer on NetMagic controller adopts libpcap to capture the packet directly.

Step 4: Determine the Internal Key Data Structure, Function Modules and Data Flow

Obviously, the most critical data structure submitted from the application support layer to the application is the packet summary structure. The MDI value calculations require the port number, packet sequence number and time stamp, and the correct transmission of packets need to set the parameters of the packet length, multiple flows detection needs to set the parameters of packet identifies, so we define the packet summary structure called Pkt_info as follows. The internal implementation of the application support layer is divided into two parts. One part is the rules management module

responsible for adding, deleting and modifying rules etc, this part will call the NMAC function to manage the rules table in NetMagic hardware; The other part is the data path function which extracts the relevant information of received packets, to generate the Pkt_info data structure which is submitted through the API to the application layer. Due to space reasons, the concrete module division not is not shown here.

```

pkt_info
{
    dst_port;// Port number

    pkt_seq;// Serial number, RTP
    protocol comes

    time_stamp;
    info_len;// The length of the data
    structure
    flow_id;// Corresponding rule ID
}

```

C. StreamMon Application Layer Design

Based on the API provided by application support layer, StreamMon application layer design is very simple. The pseudo-code of the main function and related thread are shown below.

```

1、 void Creat_flow_count_mdi_thread (void){
2、   while(1){
3、     if (check_period()==1){
4、       M=count_mdi_and_modify_list(thread_data_list);
5、       Send_MDI(pkt->flow_id,M);
6、     }
7、     if(stop_count_current_mdi == true)break;
8、   }
9、 }
10、 void Usercallback(pkt_info){
11、   ret=find_list (f_ID, list_ID_thread) ;
12、   if(Ret==0) {
13、     Creat_flow_state(pkt->flow_id);
14、     add_pkt_list(pkt,thread_data_list);
15、   }
16、 Cmd_handle(){
17、   Creat_Ctrl_socket();
18、   while(1) {
19、     if (rcv_cmd_pkt()==1) {
20、       command=parse_cmd_pkt(p);
21、       rule=gen_rule(p);
22、       switch(command){
23、         case ADD: add_rule_table(rule);
24、         case DEL: del_rule_table(rule);
25、         case MODIFY: modify_rule_table(rule);
26、       }
27、     }
28、   }
29、 main(){
30、   Register_Data_handle (&Usercallback);
31、   pthread_create(&id_message,NULL,(void
32、   *)Cmd_handle,NULL);
}

```

The StreamMon application consists of command processing thread (Cmd_handle) and data processing threads (Data_handle,). Cmd_handle will get users request information through the socket, record the user's port number, and register ID number for it, transform the request information into rule information according to the consultation with application support layer, and then call the application support layer function to issue the rule relying on the command descriptor of the requested information; Data_handle is suspended after defined, the thread define the application function according to Usercallback function provided by the application support layer, When the application support layer uploads summary packets by Usercallback, Data_handle thread will be activated, after getting the information, Data_handle thread will initialize a linked list for packets marked with each type of ID and subsequent similar packets is loaded into the corresponding data linked list, calculate the MDI value, and then start transmitting in accordance with the corresponding user port number.

D. StreamMon Achievement Results

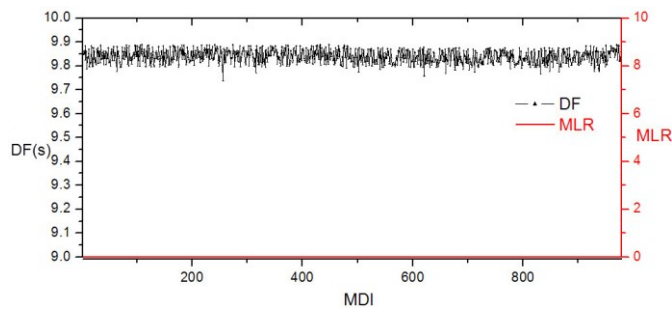


Figure 3 MDI Value Changes

Based on the NProbe UM module of NetMagic hardware and the design method discussed in 5.2.2, and 5.2.3, we designed and implemented StreamMon software. We continuously monitor the MDI value of video streaming through NetMagic for 1000 seconds monitoring, the experiment results are shown in Figure 3.

In Figure 3, the horizontal axis is the time (in seconds), the vertical ordinate on the left is the time (in seconds), the right vertical ordinate is the number. From the figure it can be observed that the DF values vary between not more than 0.1, reflecting the network jitter is very stable, MLR values are all equal to 0, indicating no packet loss happened in the network. MDI combines the two, and give a superior network video streaming quality assessment methods, showing that the network conditions that experiment tests is basically sound, and the system is operating properly.

6. Conclusion

NetMagic is gradually applied in network research. Whether it is able to minimize the researcher's using complexation is the key to determine whether NetMagic can be adopted by more and more researchers. This paper makes researches on NetMagic software development method, and proposes a three-layer model and application support layer centred design method, and takes typical application-StreamMon for example to introduce the software development process, which has important significance to NetMagic-based experiments design.

References

- [1] <http://www.Netmagic.org>
- [2] Glen Gibb, and John W. Lockwood. NetFPGA—An Open Platform for Teaching How to Build Gigabit-Rate Network Switches and Routers. IEEE Transactions on Education. 2008.
- [3] Taoli, Zhigang Sun, Chunbo Jia, Qi Su, and Myungjin Lee. Using NetMagic to Observe Fine-Grained Per-Flow Latency Measurements//Proceedings of the ACM SIGCOMM 2011, Toronto, Canada, August, 2011:466-467
- [4] Edmond W. W. Chan, Ang Chen, and Xiapu Luo, et al. TRIO: Measuring Asymmetric Capacity with Three Minimum Round-trip Times//ACM CoNEXT 2011, December 6–9 2011, Tokyo, Japan.
- [5] Cai Zhi ping. Based on active and passive measurement network measurement techniques, models and algorithms. Ph.D. Thesis.2005
- [6] Su Qi, Chen Yijiao, Jia Chunbo, Sun Zhigang, and Li Tao. Design and implementation of access and control method for NetMagic[C]. 2011, (MEC2011), Jilin, 2011.8.
- [7] Libpcap. <http://www.tcpdump.org/>