Large-scale Wire-speed Multicast Switching Structure

Based on Multipath Self-routing Switching Structure and Implemented on FPGA

Kai Cui, Hui Li*, Member,IEEE, Zhipu Zhu, Fuxing Chen
Shenzhen Eng. Lab of Converged Networks Technology. Shenzhen Key Lab of Cloud Computing Tech. & App
Shenzhen Graduate School, Peking University
Shenzhen, China.
cuikai1060@yahoo.com.cn; *corresponding author: lih64@pkusz.edu.cn

***Abstract*—Ensuring high quality of service (QoS) of multicast video stream over next generation network is a challenging issue, and how to implement the wire-speed multicast with hardware logical support in the network nodes of every hierarchy is a key solution to achieve high QoS multicast. Currently, the multicast packets are processed in this way, in which they are copied and then scheduled by ports, lastly, sent respectively. But this approach cannot ensure high QoS in real-time applications. Moreover, the traditional hardware solutions do not possess excellent large-scale scalability owning to their own bottlenecks. In this project, we have constructed a wire-speed multicast switching structure based on Multipath Self-routing Switching Structure, and implemented it on a Stratix IV FPGA chip. Additionally, we have designed the signaling system and control mechanism to support the process of self-routing and wire-speed fan-out copy of multicast packets.**

Keywords- wire-speed multicast; Multipath Self-routing Switching Structure; FPGA.

I. INTRODUCTION

According to a research report [1] published by Arbor company and University of Michigan, the video service has become the major internet application. The video stream is characterized by multicast which is achieved in two ways. One is multiple unicast software scheduling, and the other is wire-speed fan-out hardware copy. The former performs poorly in real-time property and quality of service (QoS) [2]. But the latter can guarantee the latency and also achieve excellent performance for multicast. Therefore, looking for a high QoS multicast solution which can provide hardware logical support in the network nodes of every hierarchy is a key R&D point. Currently, the multicast packets approaches cannot ensure the high QoS in real-time applications. Also, the traditional hardware solutions cannot achieve excellent large-scale scalability. In this paper, we have constructed a wire-speed multicast switching fabric based on Multipath Self-routing Switching Structure (MSSS) [3], and further implemented it on a Stratix IV FPGA chip.

The rest of the paper is organized as follows. The large-scale wire-speed multicast switching structure system is described in Section II. The hardware implementation on FPGA is given in Section III. In Section IV the real multicast stream test of the system is shown, and Section V is our conclusions.

II. THEORETICAL BASIS OF THE SYSTEM

The large-scale wire-speed multicast switching structure that we presented mainly consists of two components. One is MSSS based on group theory and the other is comparator which supports wire-speed multicast.

A. Multipath Self-routing Switching Structure

In practice, switching fabrics are usually structured by 2×2 comparators which are arranged in certain order. All of the comparators in fabric will work in cross or bar condition, according to signaling system and control mechanism, to accomplish data transfer.

In order to construct large-scale wire-speed multicast switching fabric, MSSS is presented. Such Structure is characterized by completely self-routing [3] feature, which is achieved by trace and guide [4], and high modularity and low device complexity which extremely facilitates the expansion of the network scale [5].

In the structure, parameter G indicates the group size and M indicates the quantity of groups [3]. Total number of ports is G×M. Packet loss rate caused by traffic fluctuation and sudden flow will exponentially decrease when we increase the parameter G [6]. Fig. 1 is a MSSS, in which M is 16 and G is 8, based on [:(43):(42)(31):(43):] routing network [7].

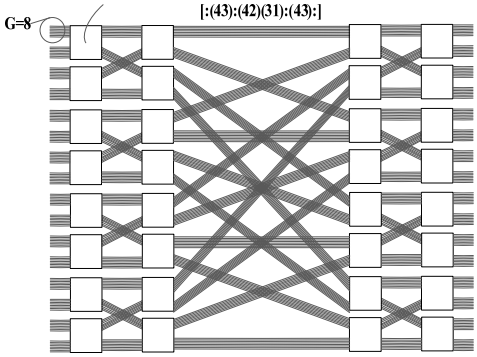

Figure 1. Multipath Self-routing Switching Structure M=16, G=8.

B. *Comparator*

Fig. 2 describes the normal 2×2 comparator which is basic element in MSSS [3].

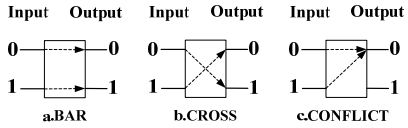

Figure 2. 2×2 comparator and its conditions

We can use two bits in-band signaling to control it [8]. The first bit A indicates activity of the input packets. When A equals 1, it means an active packet arriving. The second bit D indicates the destination of the input packet. As a result, the comparator could act upon the rule: 10<00<11. The details are shown in TABLE I.

TABLE I. TWO BITS IN-BAND SIGNALING CONTROL MECHANISM.

Conditions		**Input-1 : A1D1**		
		10	***00***	***11***
Input-0: A0D0	***10***	CONF [a]	BAR	BAR
	00	CROSS	EITHER	BAR
	11	CROSS	CROSS	CONF

a.When CONF(CONFLICT), priority decides condition.

However, when two active inputs head to the same output, we meet the conflict condition. In such condition, the priorities of two packets will be compared. The packet with higher priority would be transferred and the other one will be discarded.

When the comparator is utilized to achieve multicast, we should define and construct multicast condition, as is shown in Fig 3.

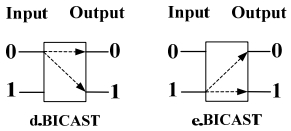

Figure 3. 2×2 comparator and its multicast condition.

To support multicast, we have updated the in-band signaling and corresponding control mechanism. As is shown in TABLE II, "B" represents the bicast packet. "I" means idle. And the number "0" and "1" indicate the destination of the unicast packets respectively [9].

TABLE II. 2×2 CONTROL MECHANISM FOR MULTICAST

Conditions		**Input-1**			
		0	***1***	***B*** [a]	***I*** [b]
Input-0	***0***	CONF	BAR	BAR	BAR
	1	CROSS	CONF	CROSS	CROSS
	B	CROSS	BAR	EITHER	BICAST
	I	CROSS	BAR	BICAST	EITHER

a.Bicast. b.Idle.

III. SYSTEM DESIGN AND IMPLEMENTION

A large-scale wire-speed multicast switching structure based on MSSS has been implemented on a Stratix IV FPGA. The parameters G and M are 8 and 4 respectively. The structure mainly comprises of 4 User Define Path (UDP) systems, each UDP serving one group of MSSS, and register system.

A. *UDP System*

The four UDP systems on FPGA are the major parts of data processing. They own three major functions. First, extract the information of packets and cells, such as total length and destination address and priority. Second, according to the information extracted, they generate various headers which we defined to assist the data processing, such as routing control header and splitter header and cell assemble header. Third, accomplish the switching, including constructing the switching fabric and cells compare.

1) *The Structure of UDP*

The four UDP systems are implemented on the FPGA chip which is connected to four external PHY chips.

Fig. 4 describes the structure of UDP which consists of seven sub-modules. The arrows indicate the direction of data flow.

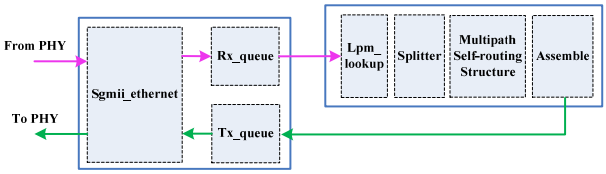

Figure 4. User Define Path.

The functions of each sub-module are as follows.

a) Sgmii_ethernet: It is the interface between UDP and external PHY chip. We utilized the Triple-Speed Ethernet (TSE) IP core supported by Altera to construct it.

b) Rx_queue: The input of this sub-module is standard Ethernet frames. It extracts the information of the frames, such as length and address, and generates the splitter header which helping the splitter to split each frame into cells in certain length.

c) Lpm_lookup: Look up the routing table and generates the lpm header including the destnation address and priority.

d) Splitter: Split each frame into cells in certain length and generate the routing control header, which is used for cells compare in the comparator of MSSS, and cells assemble header, which is used for assembling the cells into standard Ethernet frame after the switching.

e) Multipath Self-routing Structure: It implements the MSSS. Cells switching takes place in this sub-module.

f) Assemble: Assemble the arriving cells, coming from MSSS, back to frame and generate the starting index header to show the very beginning of each frame.

g) Tx_queue: Restore the standard Ethernet frames completely and send the frames to the Sgmii_ethernet.

2) Data Processing in UDP

Fig. 5 describes frame format in the UDP. In the figure, the tables demonstrate the frame format of the very place where brown arrows deriving from.

Moreover, as is shown in Fig. 5, the data width in UDP is 8 bits. We extra added CTRL signal of 2 bits to assist in data identification and processing. The control and data signals will transmit in parallel. When "11" is detected, it means current data is the first byte of a fresh new frame. And "01" means the header information generated by UDP. "00" means the payload of standard Ethernet frame and "10" means the last byte of a complete frame.

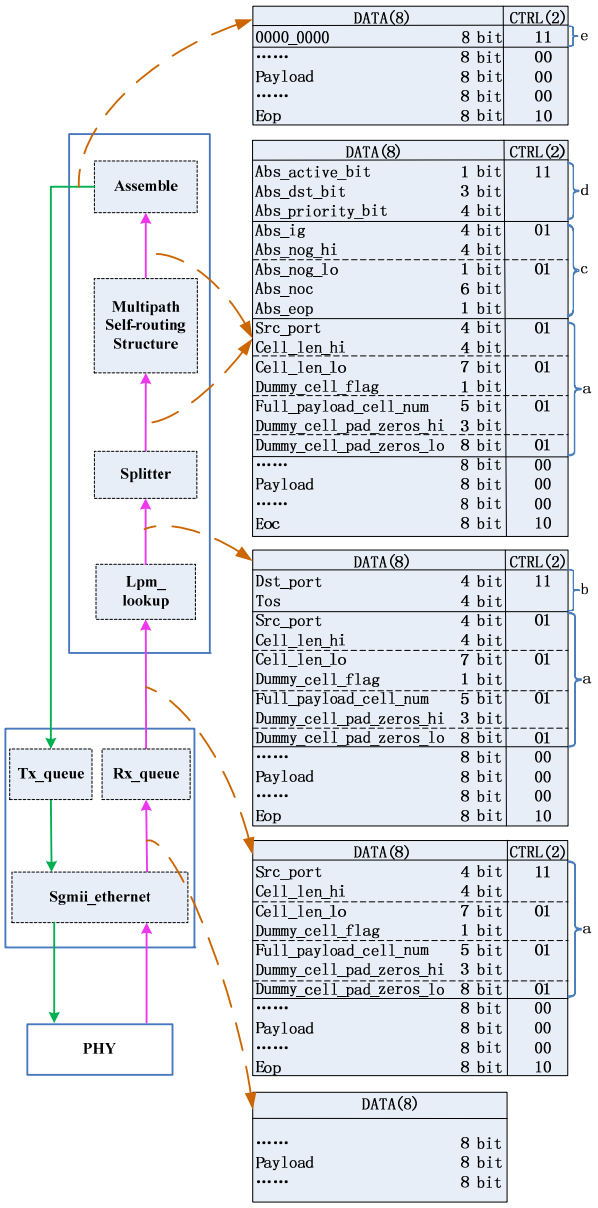

Figure 5. Data processing in UDP.

According to the discuss above, we know that the UDP would extract the information of fames to generate five kinds of headers which are labeled as "a", "b", "c", "d" and "e" in Fig. 5. They are splitter header, LPM header, cell assemble header, routing control header, and starting label header respectively. The headers will be attached in front of the former frame.

a) Splitter header: It is generated by the Rx_queue whose input is standard Ethernet frame. Splitter header contains four bytes information. Src_port is the source address of the current frame. Cell_len shows the length of the cell which is set to 128 bytes. Dummy_cell_flag will be set to high when the current frame cannot be split to a certain number of cells exactly. When a frame remains a tail which is less than 128 bytes after being split to several cells, we would expand it to 128 byes with dummy bytes which are all 0x00. And the number of dummy bytes is recorded in Dummy_cell_pad_zeros.

The Multipath Self-routing Structure is the very place where the switching proceeds. Considering the efficiency and feasibility, we use cells in certain length to complete the switching rather than frames of various lengths.

b) LPM header: Generated by Lpm_lookup and contains only one byte. Dst_port is destnation port of the frame and Tos is the priority information which is extracted from the Ethernet frame.

c) Cell assemble header: The splitter utilizes the splitter header to split every frame into cells, with the same length, and additionally generates the cell assemble header for each cell. Abs_ig is the input group number of cell. Abs_nog gives the port number in the group which includes 8 ports in this system, as a result of G = 8. Abs_noc is a significant serial number which mirrors the actual position of the cell in the original frame. We will make use of it to recover the original frame after the switching of cells. And abs_eop will be set to high when the current cell is the tail of a frame.

d) Routing control header: It is the signaling which is generated by Splitter and utilized to control the comparators in MSSS. Abs_active_bit show the activity of the cell and Abs_dst_bit gives the destinations. When conflict, the Abs_priority_bit will be taken into account to decide which cell would be transmit.

e) Starting label header: Created by Assemble to show Tx_queue the very beginning of the frame.

When the frame is passed to the PHY by Sgmii_thernet, the switching procedure is completely finished.

B. Register system

Another major component of the large-scale wire-speed multicast switching structure on FPGA is the register system which has two functions. On one hand it configures the sub-modules in UDP through software registers; on the other it extracts the internal signals of the UDP to help us debug the system, through hardware registers.

There exist two kinds of registers, software register and hardware register. Software registers can be written to configure the modules on FPGA. And hardware registers contain vital information of modules such as state of Finite State Machine (FSM), which is greatly helpful for system debug.

We adopted the pipeline architecture to design our register system. Registers are serially connected by specific interface. Every register only responses to the requests which belonging to it and passes on the others to the next module. Compared to the star architecture system, it is much more convenient when we add another module into system, only updating the address map.

The register system is constructed in Qsys development environment which is attached in Quartus II. We utilized Jtag_Avalon_Master_Bridge and made the register interface for every sub module based on Avalon Bus Specification. All the read and write operations are realized by using Tcl script.

Through the register system, we can use a computer to exchange signals with the large-scale wire-speed multicast switching structure system on FPGA.

IV. SYSTEM TEST WITH REAL MULTICAST STREAM

After implemented the whole system on FPGA, we have tested it with real network stream.

Our primary network test instrument is IXIA 400T network tester. It provides various test modules that support 10/100/1000 Ethernet standard and can generate and count and capture all kinds of streams. It not only gives detailed statistical data but also can offer every byte data of frames.

Now, the system has passed all the normal functional tests including unicast and multicast. Next step, much more complicated tests could be implemented to figure out the real performance of large-scale wire-speed multicast switching structure in all kinds of network environment.

V. CONCLUSIONS

This paper has structured multicast comparator and combined with the Multipath Self-routing Switching Structure to construct the large-scale wire-speed multicast switching structure. Moreover, we have implemented the system on a Stratix IV FPGA and tested it with real network stream.

During the process of constructing the whole system on FPGA, we first devised the entire architecture, and then analyzed and designed every sub-modules and the cooperative relationship between them, and then the signaling system and control mechanism to support the process of self-routing and wire-speed fan-out copy, at last the system has been tested with the real network stream.

Currently, the ports number of the system we constructed on FPGA is relatively limited. Further work could be done to increase the scale of the structure and improve the efficiency of signaling system and control mechanism which are appropriate for the applications of large-scale wire-speed multicast.

ACKNOWLEDGMENT

The work is supported by National Basic Research Program of China (973 Program) (No.2012CB315904), National Natural Science Foundation of China (No.61179028), Natural Science Foundation of Guangdong Province (No.201101000923), Basic Research of Shenzhen (No.201104210120A, No.201005260234A) and Shenzhen Industry (No. 201006110044A).

REFERENCES

[1] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, Farnam Jahanian; —Internet Inter-Domain Traffic, ACM SIGCOMM 2010 ;

[2] A. S. Tanenbaum, Computer Networks, 4thEdition, Prentice Hall, 2003.

[3] Hui Li, Wei He, Xi CHEN, Peng Yi, Binqiang Wang, “Multi-path Self-routing Switching Structure by Interconnection of Multistage Sorting Concentrators”, IEEE CHINACOM2007, Aug.2007, Shanghai ;

[4] Li S Y R. Algebraic switching theory and broadband applications. Academic Press, 2001 ;

[5] Li H, Li S Y R. Layout complexity of bit-permuting exchange in multi-stage interconnection networks[M]. Switching networks: recent advances, Kluwer Academic Publishers, Boston, USA, 2001, 259-276.

[6] He W, Li H, Wang B, et al. A Load-Balanced Multipath Self-routing Switching Structure by Concentrators[C]. IEEE ICC 2008 ;

[7] Li H, Li S Y R. On the Complexity of Concentrators and Multi-stage Interconnection Networks in Switching System[D]. The Chinese University of Hong Kong, 2011.

[8] Li S Y R. Unified algebraic theory of sorting, routing, multicasting, and concentration networks[J]. Communications, IEEE Transactions on. 2010, 58(1): 247-256.

[9] Li S Y R, Li H, Koo G M. Fast knockout algorithm for self-route concentration[J]. Computer Communications. 1999, 22(17): 1574-1584.