Submit to: Workshop on Computer Networks

Design of a Network Layer Protocol Transformer Between IP and ATM

Tzeng-Yi Lin, Yeu-Horng Shiau, Shiann-Rong Kuang, and Jer-Min Jou

Abstract

Recently, network grows explosively and its scope extends rapidly so that the interconnection and communication between each type of network are more important. This paper presents a technique and its hardware implementation for network layer protocol transform between IP and ATM. Experimental results show that the proposed architecture can quickly transform the data format between IP and ATM network to meet the real-time requirement.

Key words: TCP/IP, ATM UTOPIA, Network Layer, IP over ATM

Tzeng-Yi Lin, (the contact author)

Current affiliation: Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, R. O. C.

Postal address: ASIC LAB, EE 10F, NCKU, NO.1, Ta-Hsueh Road, Tainan, 701 Taiwan

E-mail address: vitamin@j92a21.ee.ncku.edu.tw

Telephone number: 06-2757575-62431-821

Yeu-Horng Shiau,

Current affiliation: Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, R. O. C.

Postal address: ASIC LAB, EE 10F, NCKU, NO.1, Ta-Hsueh Road, Tainan, 701 Taiwan

E-mail address: huh@j92a21.ee.ncku.edu.tw

Telephone number: 06-2757575-62431-821

Shiann-Rong Kuang,

Current affiliation: Department of Electronic Engineering, Southern Taiwan University of Technology, Tainan, Taiwan, ROC

Postal address: as above

E-mail address: kuangsr@mail.stut.edu.tw

Telephone number: 06-2533131-3131-232

Jer-Min Jou

Current affiliation: Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, R. O. C.

Postal address: ASIC LAB, EE 10F, NCKU, NO.1, Ta-Hsueh Road, Tainan, 701 Taiwan

E-mail address: jou@j92a21.ee.ncku.edu.tw

Telephone number: 06-2757575-62365

1. Introduction

ATM network communication technology includes the advantage of dynamic bandwidth location, low transmission delay, bandwidth and quality of service guarantee. It can provide different transmission medium such as optical fiber, twisted pair, and coaxial cable, so that it can be applied to LAN (Local Area Networks), MAN (Metropolitan Area Network), and WAN (Wide Area Network). By the reason of above description, ATM is regarded as one of the most important technology in the future network, and it is almost possible to play the role of the backbone of Wide Area Network. And Broadband—Integrated Service Digital Networks will also build ATM network.

However, the most important communication protocol in the internet is IP at present. It is entirely different from the type of ATM. IP is connectionless; but ATM is connection—oriented. Hence it is a very important subject about how to transmit IP packet in ATM network. Although ATM has many advantages, the network structure advance also must consider existed network device. Therefore, when we want to use ATM backbone to connect WAN and traditional LAN, it is necessary to design hardware architecture and software system of the edge router.

The main concern in this paper is the data format transformation between IP and ATM. First, we must realize the data format transmitted in IP and ATM network. The IP packet length is not fixed, the minimum is 64 bytes and the maximum is 1492 bytes. The ATM packet is called a cell, and it has a fixed length of 48 bytes. In addition to data length, another object that we must realize is the header because the protocol behavior and data relay both depend on the content of the header. The IP and ATM headers are shown in Fig. 1 and Fig. 2, respectively.

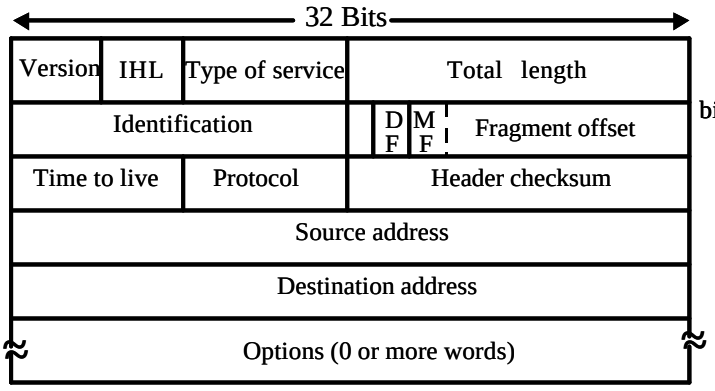


Figure 1 IP header

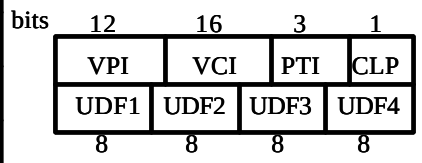


Figure 2 ATM header

2. System Architecture

The system architecture of the proposed protocol transformer is shown in Fig. 3. The following paragraphs describe its function in detail.

When it receives the first 4 bytes of an IP packet from the Ethernet Network Interface Card, the first work is checking that whether its version is 4 and whether its header length is equal to 5 or greater than 5. If one of the two conditions doesn't satisfy, the IP packet must be discarded. If both conditions are satisfied, the remainder of the IP packet is received continually and its checksum is calculated at the same time. Until the packet header is received completely, if its checksum is wrong, the packet must be discarded; otherwise the receiving action continues until the packet end is being received.

It must be decided that whether the IP packet needs to be fragmented into ATM cell. If it need not be fragmented, during the receiving process in "IP to IP" block it inquires into the output port/next hop from the routing table according its destination address, then is transmitted to the IP switch fabric. If it needs to be fragmented, waiting for ATM connection during receiving. We suppose that the router or switch maintains an IP-ATM connection mapping table, so the VPI、VCI and port obtained after connecting completely are part of input signals. When the packet is received

completely and enough information is got, the "IP to ATM" block starts to fragment and transmit.

When receiving cells from the ATM UTOPIA interface, it is not necessary to check them because error detection has proceeded in Physical Layer. It must also be decided that whether the cells need to be reassembled in order to transform into IP packets or directly switched to the output port. In the former, "ATM to ATM" block takes the VPI in the ATM cell header as the index to inquire VPI mapping table, in order to decide output port and new VPI, and transmit the cell to the ATM switch after the rest has been received completely. In the later, instead of providing queue for every VPI to reassemble the cells, "ATM to IP" block uses the management memory of VPI reassemble information to store reassembling status of individual VPI and the memory address link between reassembled cells. This can save memory resource. The IP packet that is reassembled completely will be transmitted to corresponding output port according to which VPI it is reassembled.

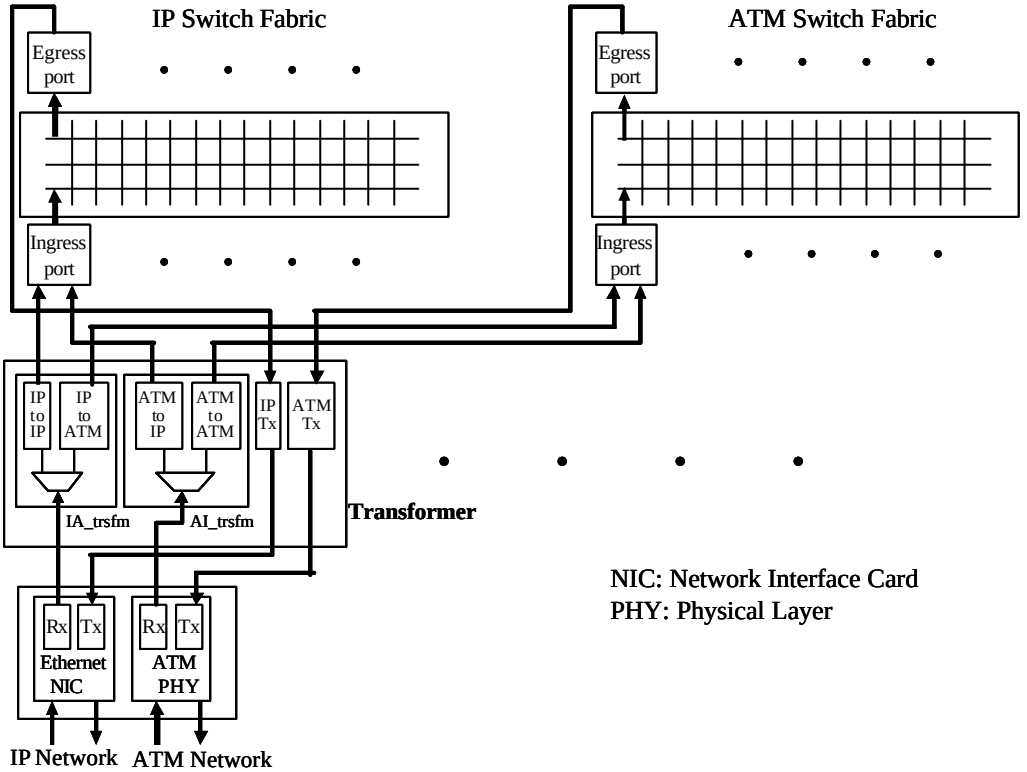


Figure 3 System architecture

3. Implementation

The main work of Fig. 3. can be classified by function into four parts: packet receiving process, cell receiving process, packet transmitting process, and cell transmitting process. The modules to perform these processes will be explained in detail in the following subsections.

3.1 Packet Inbound Process

The first module is IA_trsfm module shown in Fig. 4, including “IP to IP” block and “IP to ATM” block in Fig. 3. When this module and Ethernet network interface card are both ready, the packet is written to the memory, and the counter that generates receiving addresses of the memory starts to count. At the same time, this module checks the last byte of the first received 4 bytes of the packet (data width is 4 bytes). The last 4 bits of the checked byte are IP version, and its value must be 4. The first 4 bits of the checked byte are packet header length, and per unit equals to 4 bytes. An IP packet header length is at least 20 bytes, so the value of the first 4 bits must be equal to or greater than 5. The counter that generates receiving addresses of the memory continues counting if the two conditions described above are both satisfied. When data is written to the memory, it simultaneously is sent to IP checksum block. Until the packet header is received completely, IP checksum block generates the check result. If the result is correct, the packet is received continually; otherwise the receiving address of the memory returns to the state before receiving.

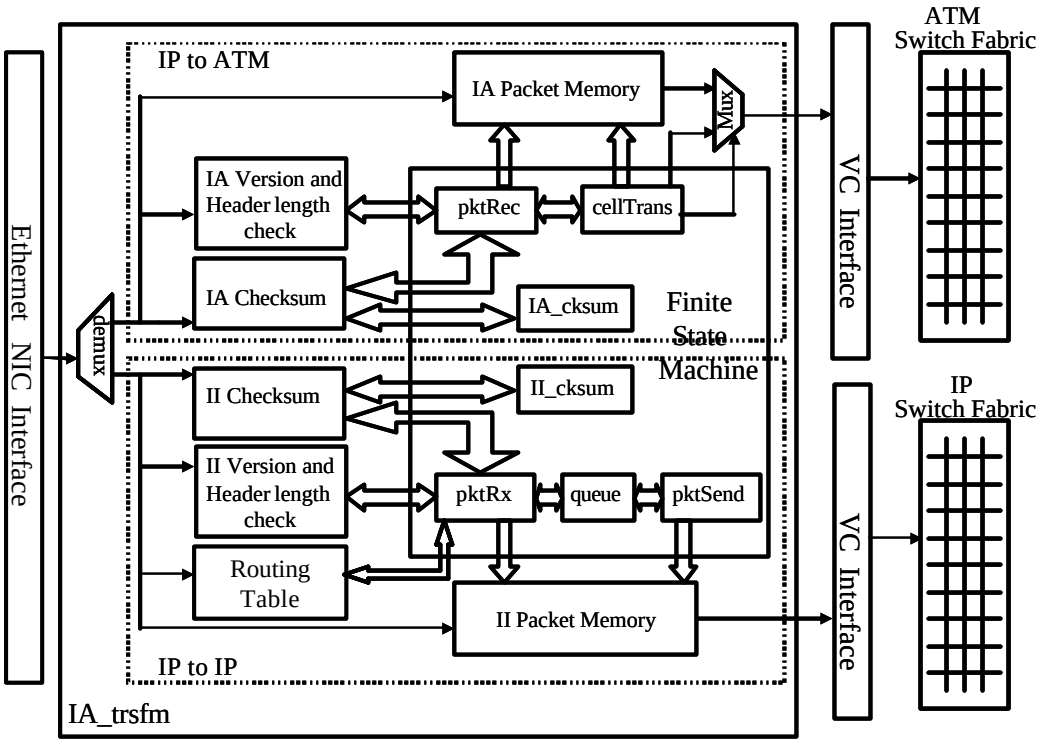


Figure 4 IA_trsfm module

Checksum

Because data width is 32 bits, so we use 32-bit algorithm for checksum. The detailed checksum block is shown in Fig. 5. During receiving the packet, IP checksum block simultaneously takes 4 bytes in the header as a unit to accumulate the checksum. When the header is received completely, the block goes into the next state and then checks whether the last accumulative result overflows. If the result overflows, the value of the last 32 bits and the first 32 bits in the 64-bit sum will be added. This action is repeated until no overflow, then this block goes into the next state. In this state, IP checksum block checks whether the value of the last 16 bits in the 32-bits result is zero. If the value is not zero, as the previous state, the value of the last 16 bits and first 16 bits in the 32-bit result will be added and this action is repeated until the value of the last 16 bits is zero. Finally, IP checksum block checks whether the first 16 bits are 16' hFFFF. If they are 16' hFFFF, the packet is valid; otherwise the receiving

address of the memory returns to the state before receiving as the packet is discarded.

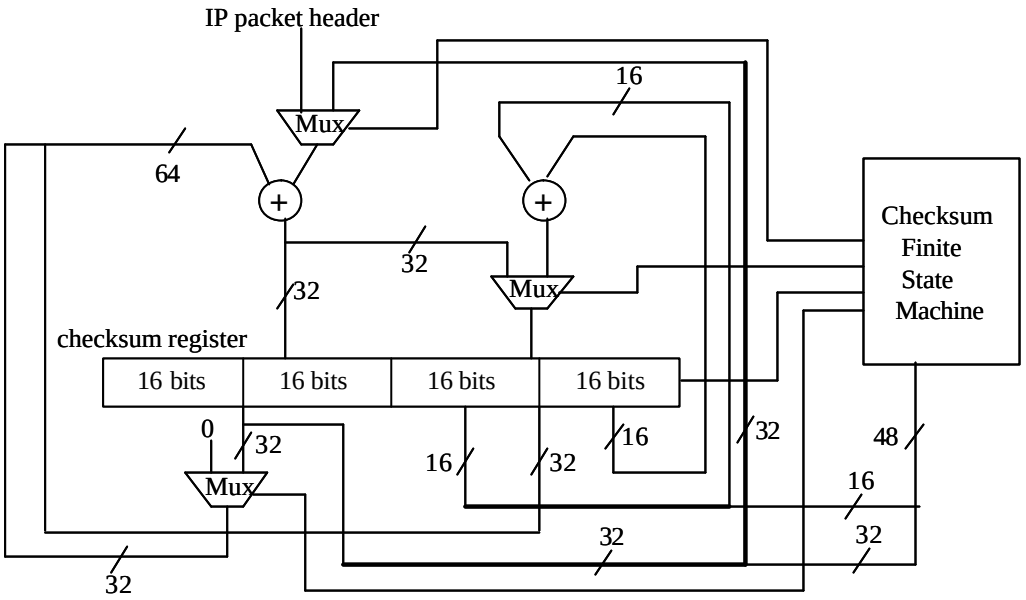


Figure 5 Checksum block

Routing Table

If the IP packet need not be fragmented, it is essential to inquire Routing Table to get next output port. Routing Table shown in Fig. 6 includes three memories that has 256 words (word width is 4 bits). The initial contents of them are all wrote in advance. The operation of inquiry is as follows (based on original intention of verification, this block simply uses the first 24 bits of the IP address as the index)： the first 24 bits of the IP address are separated into three 8-bit, each of them is the individual read-address of the three memories described above. The width of the memory is 4 bits, the first bit represents that whether the last 3 bits are the final routing result (this similarly means the Longest Prefix Match), and the last 3 bits represent the output ports. The Next Hop outputted is stored in one queue, and is taken out when this packet is transmitted out.

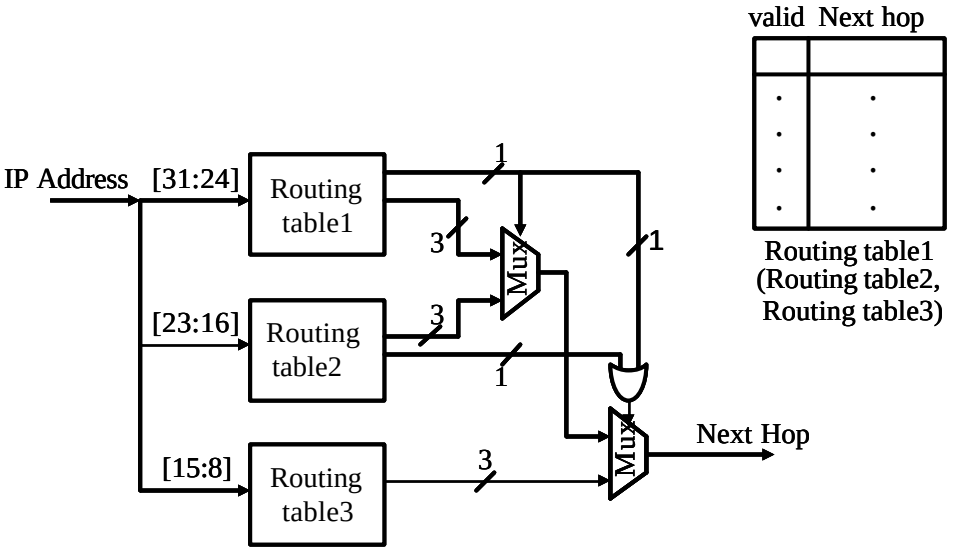


Figure 6 Routing Table

Packet Fragmenting (Controlled by FSM)

While the packet is received, its length would be stored. And, as the packet is transmitted, the length is used for the judgment that whether the packet fragmenting is over. Since the packet would be segmented into ATM cells, the finite state machine takes 48 bytes as a unit, adds a cell header for each unit, and then transmits them. Because the data width is 32 bits, the ATM cell length is a multiple of 4 bytes (52 bytes or 56 bytes). We adopt the mode of 56 bytes. The first 4 bytes in the header are composed of VPI, VCI, PT, and CLP, and the last 4 bytes are UDF (User Define Field). Due to the ability of the ATM cell reassembling at end, UDF is used to judge that whether it is the last cell in the fragmented IP packet. If UDF is 32'hFFFFFFFF, this cell is not the packet end; else if UDF is 32'h00000000, it is the packet end. Process described above is illustrated in Fig. 7.

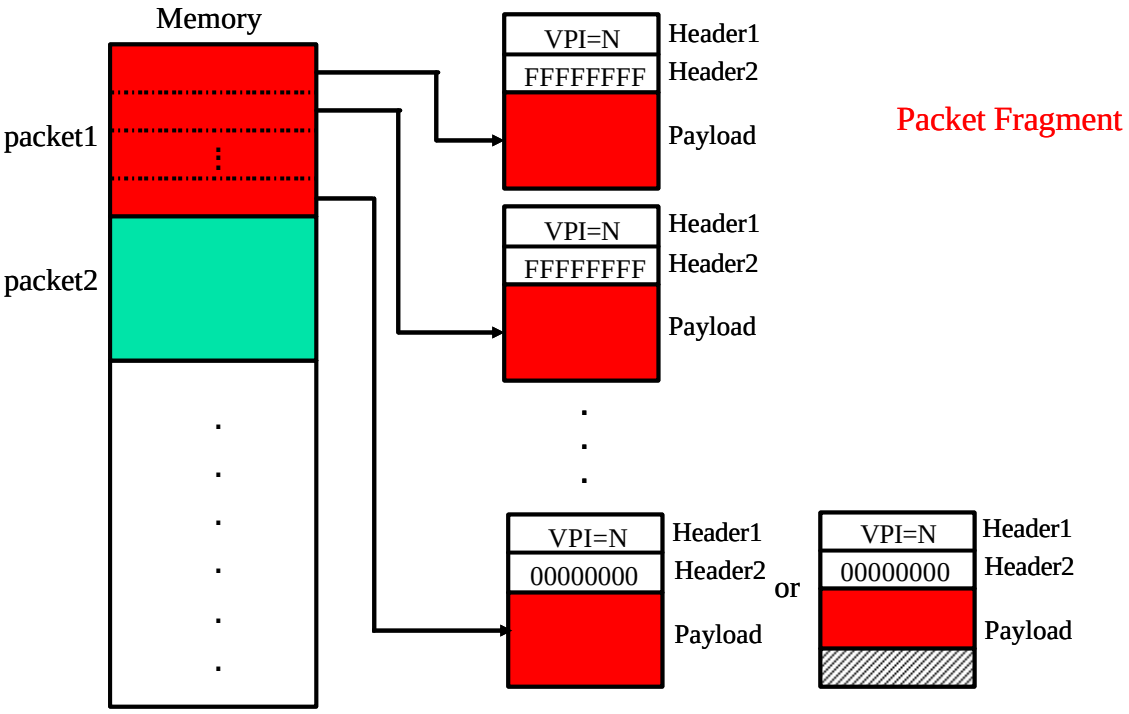


Figure 7 Packet fragment example

3.2 Cell Inbound Process

The second module is AI_trsfm module shown in Fig. 8, including "ATM to IP" block and "ATM to ATM" block in Fig. 3. Between physical layer and ATM layer is UTOPIA interface. If the ATM cell received from this interface need not be reassembled, "ATM to ATM" block inquires VPI table and then transmits it to ATM switch fabric.

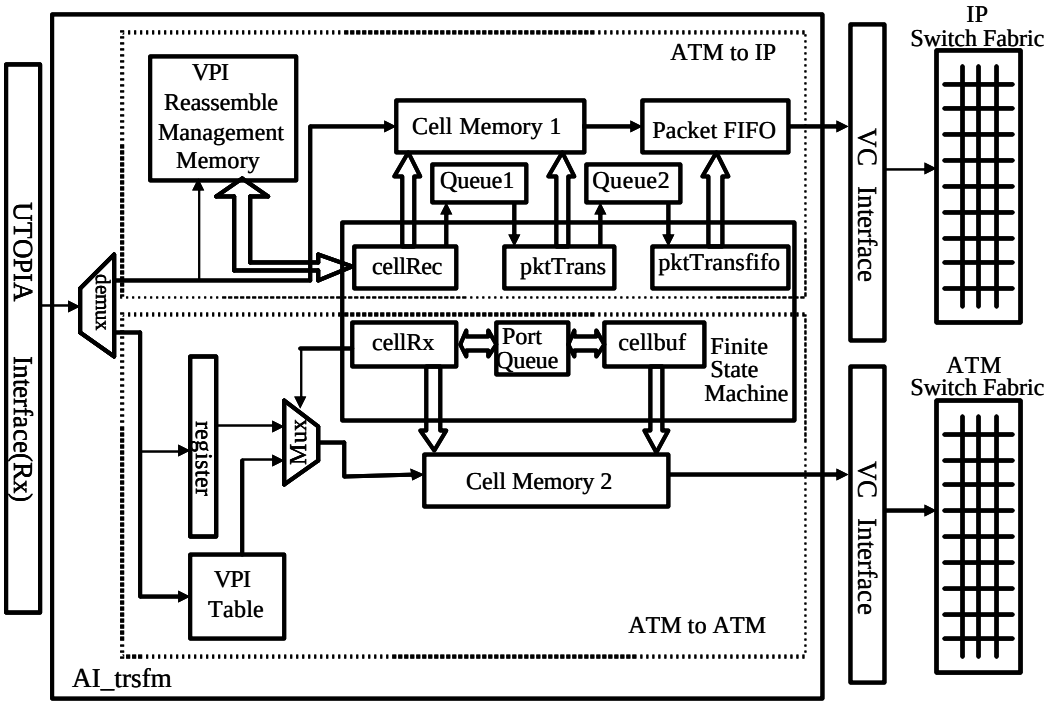


Figure 8 AI_trsfm module

Cell Reassembling

(1) Receiving

The cell reassembling relies on a memory that manages VPI reassembling information. While the cell is received, VPI in the header would be the index to read out its reassembling information stored in the memory. The width of the memory of VPI management is 17 bits, the MSB represents if this cell is the first reassembling cell. If the MSB is 0, this cell is the first cell, and MSB would be changed into 1 and written into the memory at next clock to mean that each following cell received with this VPI is not the first cell about this VPI. The 8 bits—[15:8] in the memory width represent the start address of the first cell of some VPI. So while the first 4 bytes of the first cell arrives, these 8 bits read out must be replaced with the value of the counter generating receiving addresses of the memory at that time and then written into the memory. This value then is maintained until the cell reassembling terminates. The last 8 bits—[7:0] in the memory width represent the address next to the address of the last 4 bytes of the preceding received cell. If the received data is not the first cell with this VPI, at that

time the receiving address generated from the counter should be written into the cell memory location that its address is the value of the 8 bits—[7:0] in the width of the VPI management memory, for the purpose of being the information of link list.

While the UDF in the incoming ATM cell header is 32’h00000000, it is the last reassembling cell, and the MSB in the width of the memory of VPI reassembling information should be changed into 0 to mean that the next received cell with this VPI is the first cell of the next IP packet. Process described above is illustrated in Fig. 9.

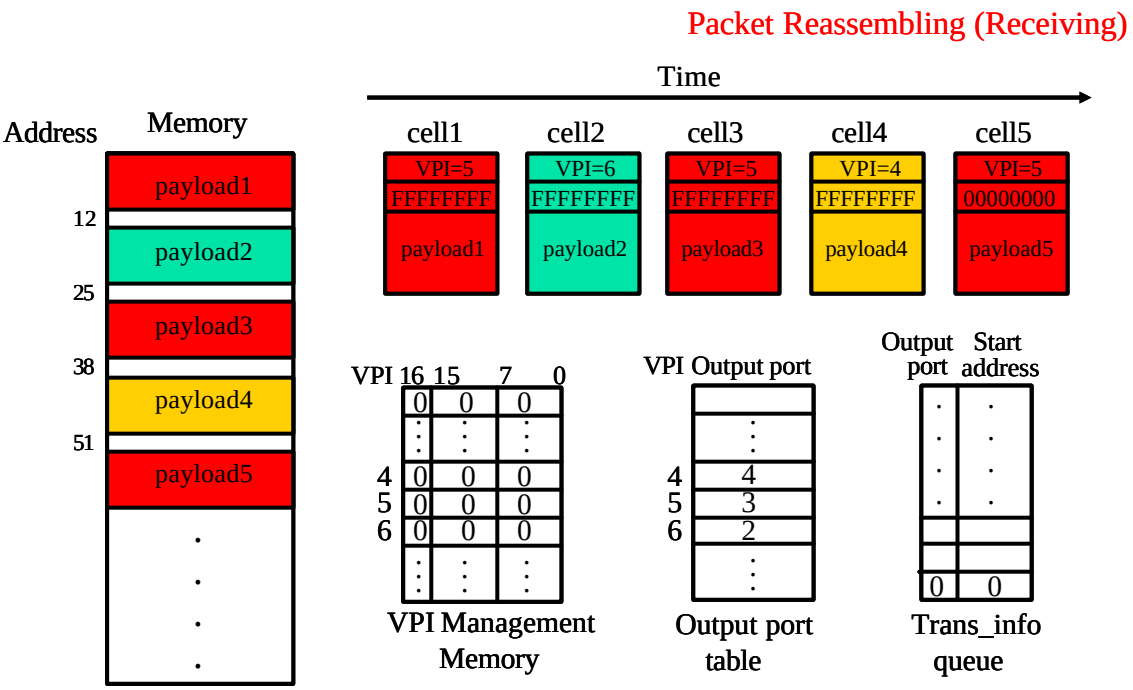


Figure 9(a) Cell receiving

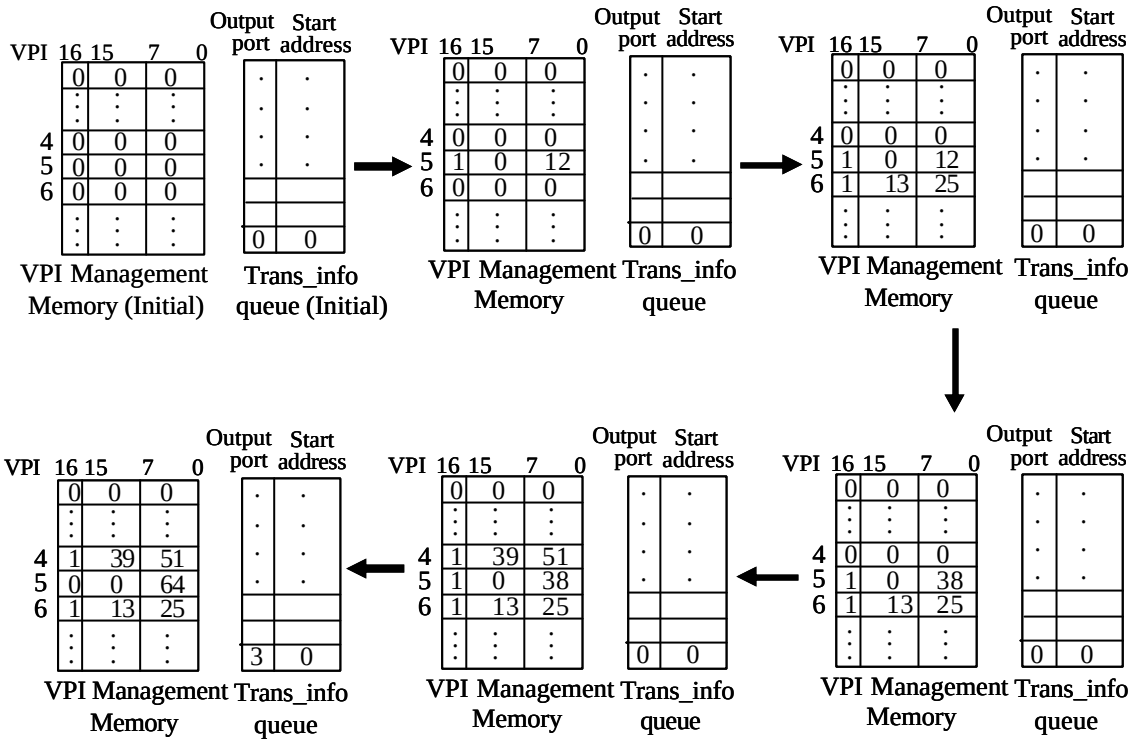


Figure 9(b) Variation of VPI Management Memory and Trans_info queue

(2) Transmitting

As some packet is reassembled completely, its start address stored in the VPI management memory would be queued, read out in sequence, and then the packet starts being transmitted to a FIFO. Why is not the packet directly sent out and first delivered to a FIFO instead? Because the transmission between reassembled cells need to take one clock time to read out the next cell access address in the link list of the former cell end, the direct transmission will cause IP switch received incontinually. There is no problem about this if first store the packet in the FIFO and then transmit it. Process of transmitting to packet FIFO is illustrated in Fig. 10.

Packet Reassembling (Transmitting to Packet FIFO)

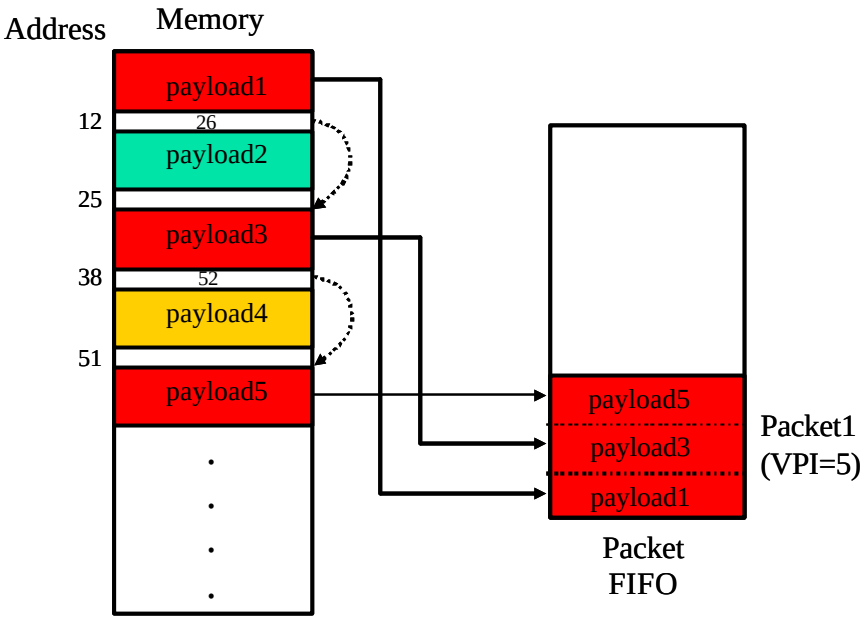


Figure 10 Transmitting a packet that is reassembled completely to FIFO

VPI Table Block

The main work of the "VPI Table" block is to update the VPI. Its major structure is a memory with the width of 8 bits. The VPI in the incoming cell header is the read-address for this table. The last 4 bits in the content read out are new VPI, and the first 4 bits are output port. Fig. 11 shows an example for VPI table.

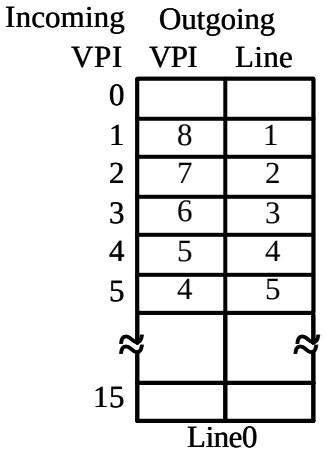


Figure 11 VPI table

3.3 Packet Outbound Process

The third module shown in Fig. 12 is "IP Tx" block in Fig. 3. As the packet would

be sent to the network, the checksum has to be reset to zero and the TTL (time to live) ought to subtract one, then the checksum must be calculated again and put in the checksum field. However, if designing this circuit according to the procedure, the efficiency of packet delivering will be decreased. So we design another equivalent circuit, its action is described as follows: First, calculate the 1's complement of the received checksum, then the sum of the original packet header can be obtained. Next, consider that if taking 16 bits as a unit to sum up the header, the action that TTL subtracts one means that the header sum subtracts 256. So after the original sum subtracts 256, calculate its 1's complement and the new checksum can be acquired.

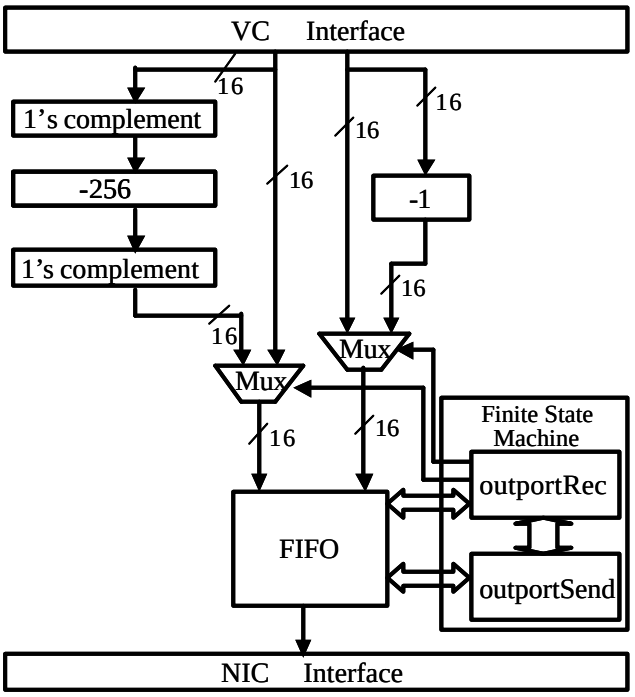
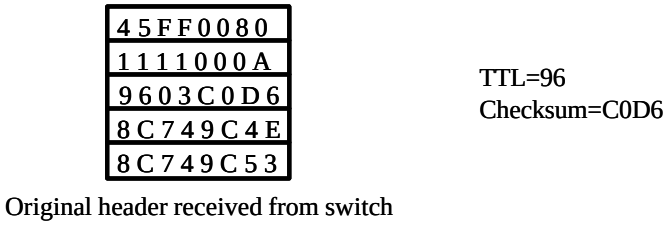


Figure 12 "IP Tx" module

TTL decrease and Checksum Update

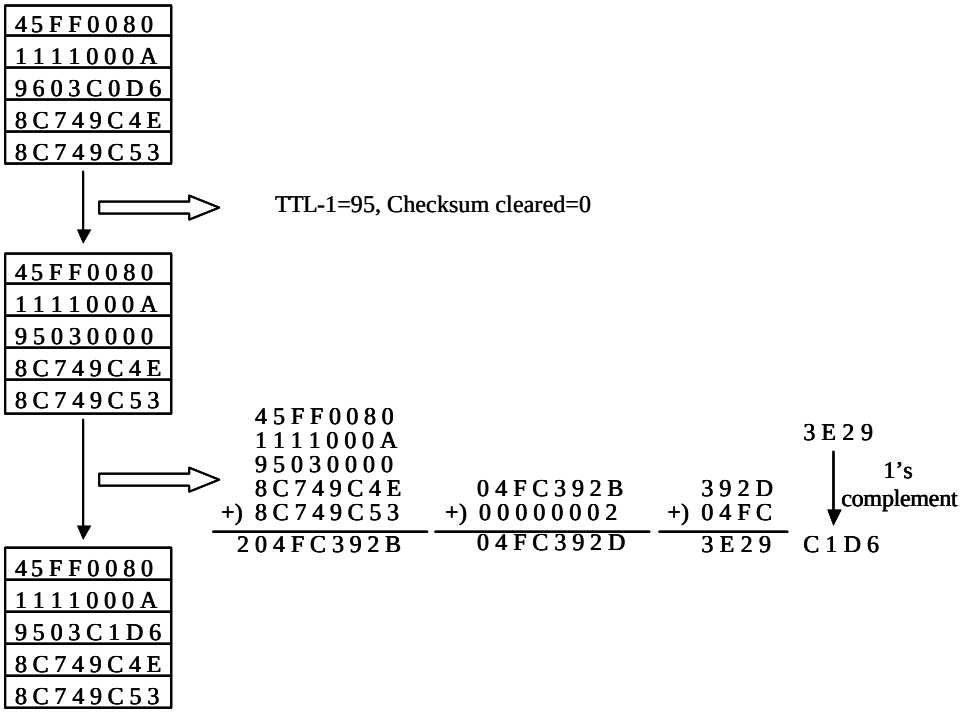
The actions of checksum update and TTL decrease complete during the former described process—packet receiving from IP switch. After packet receiving from IP switch begins, the third coming data includes checksum and TTL, so at this clock they must be changed. Why does the data flow in Figure 6.1 replace the normal process that consists of clearing original checksum, decreasing TTL, and then recalculating

checksum? We illustrate the reason with normal and improved processes of a standard 20-byte header in the following example (Note that the header content values are presented in hexadecimal):

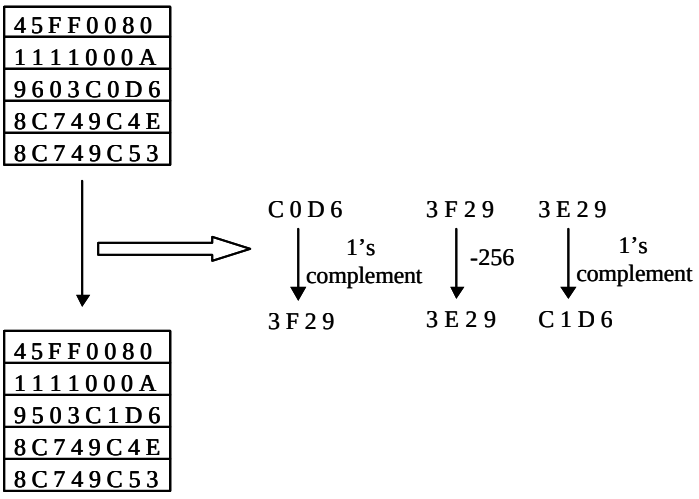


Original header received from switch

Normal process:



Improved process:



3.4 Cell Transmitting Process

The fourth module is the "ATM Tx" block in Fig. 3, and it is the simplest module. Its task just receives cells from the ATM switch and transmits cells to physical layer. Fig. 13 presents the "ATM Tx" module.

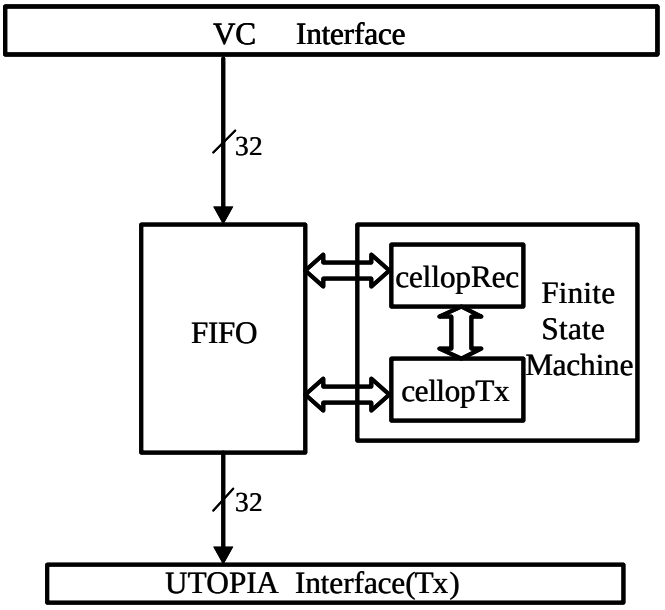


Figure 13 "ATM Tx" module

4. Interface and Experiment results

This section discusses the interfaces between the four modules and other components in the system architecture. The IA_trsfm module receives packets through the NIC interface, and transmits packets or cells through Virtual Component Interface (VCI) to switches. Note that the VCI is an on-chip bus standard for SOC. The AI_trsfm module receives cells through UTOPIA interface established by ATM Forum and transmits cells or packets through VCI to switches. The "IP Tx" and "ATM Tx" modules receive packets or cells from VCI, and transmit them through NIC and UTOPIA interface.

The four modules in section 3 are implemented by using Verilog code. The simulation results and design summaries for the four modules are listed in Table 1 to

Table 4.

5. Reference

[1]. Andrew S. Tanenbaum, Computer Networks, Third Edition, Prentice-Hall, 1996.

[2]. http://www.atmforum.com, About ATM Technology, ATM standardization.

[3]. ATM Forum Technical Committee, af-phy-0017.000, "UTOPIA Specification, Level 1, Version 2.01," March 21, 1994.

[4]. ATM Forum Technical Committee, af-phy-0039.000, "Utopia Level 2, Version 1.0", June 1995.

[5]. ATM Forum Technical Committee, "UTOPIA 3 Physical Layer Interface," November 1999.

[6]. J. Touch, B. Parham, "Implementing the Internet Checksum in Hardware," IETF RFC 1936, April 1996.

[7]. R. Braden, D. Borman, and C. Partridge, "Computing the Internet Checksum," IETF RFC 1071, September 1988.

[8]. Gray R. Wright and W. Richard Stevens, TCP/IP Illustrated Volume 2: The Implementation, Addison-Wesley, 1995.

[9]. Pankaj Gupta, Steven Lin, and Nick McKeown, "Routing Lookups in Hardware at Memory Access Speeds," Computer Systems Laboratory, Stanford University Stanford, CA 94305-9030.

[10]. Andreas Moestedt and Peter Sjödin, "IP Address Lookup in Hardware for High-Speed Routing," Swedish Institute of Computer Science, P.O. Box 1263, SE-164 29 KISTA, Sweden.

[11]. Peter Newman, Greg Minshall, Tom Lyon, and Larry Huston, Ipsilon Networks Inc, "IP Switching and Gigabit Routers," IEEE Communications Magazine, January 1997.

[12]. VSI Alliance, On-Chip Bus Development Working Group, "Virtual Component Interface Standard Version 2(OCB 2 2.0)," April 2001.

[13]. James Aweya, "IP Router Architectures: An Overview," Nortel Networks, Ottawa, Canada, K1Y 4H7.

[14]. O. G. Koufopavlou, A. N. Tantawy, and M. Zitterbart, "Analysis of TCP/IP for High Performance Parallel Implementations," IBM Research Division, T. J. Watson Research Center, P. O. Box 704, Yorktown Heights, NY 10598.

[15]. Nick McKeown, "High Performance Switching and Routing," http://www.stanford.edu/~nickm.

Table1 AI_trsfm block

Resource	No. of used	Max Available	% used
Number of Slices:	2,026	12,288	16%
Number of Slice Flip Flops:	2,294	24,576	9%
Total Number 4 input LUTs:	3,898	24,576	15%
Number used as LUTs:	3,896		
Number used as a route-thru:	2		
Number of bonded IOBs:	143	404	35%
Number of Block RAMs:	7	32	21%
Number of GCLKs:	1	4	25%
Number of GCLKIOBs:	1	4	25%
Total equivalent gate count for design: 159,797			
Additional JTAG gate count for IOBs: 6,912			

device xcv1000, package BG560 , speed -4
Design statistics:
Minimum period: 31.551ns (Maximum frequency: 31.695MHz)
Maximum combinational path delay: 35.685ns
Maximum net delay: 21.633ns

Table2 IA_trsfm block

Resource	No. of used	Max Available	% used
Number of Slices:	1,035	12,288	8%
Number of Slice Flip Flops:	447	24,576	1%
Total Number 4 input LUTs:	1,938	24,576	7%
Number used as LUTs:	1933		
Number used as a route-thru:	5		
Number of bonded IOBs:	192	404	47%
Number of Block RAMs:	7	32	21%
Number of GCLKs:	1	4	25%
Number of GCLKIOBs:	1	4	25%
Total equivalent gate count for design: 131146			
Additional JTAG gate count for IOBs: 9264			

device xcv1000, package BG560 , speed -4
Design statistics:
Minimum period: 74.114ns (Maximum frequency: 13.493MHz)
Maximum net delay: 16.392ns

Table3 IP Tx block

Resource	No. of used	Max Available	% used
Number of Slices:	1776	12,288	14%
Number of Slice Flip Flops:	2157	24,576	8%
Total Number 4 input LUTs:	3533	24,576	14%
Number used as LUTs:	3532		
Number used as a route-thru:	1		
Number of bonded IOBs:	70	404	17%
Number of GCLKs:	1	4	25%
Number of GCLKIOBs:	1	4	25%
Total equivalent gate count for design: 41280			
Additional JTAG gate count for IOBs: 3408			

device xcv1000, package BG560 , speed -4
Design statistics:
Minimum period: 33.537ns (Maximum frequency: 29.818MHz)
Maximum combinational path delay: 44.517ns
Maximum net delay: 31.049ns

Table4 ATM Tx block

Resource	No. of used	Max Available	% used
Number of Slices:	2355	12,288	9%
Number of Slice Flip Flops:	2078	24,576	8%
Total Number 4 input LUTs:	4703	24,576	19%
Number of bonded IOBs:	71	404	17%
Number of GCLKs:	1	4	25%
Number of GCLKIOBs:	1	4	25%
Total equivalent gate count for design: 44842			
Additional JTAG gate count for IOBs: 3456			

device xcv1000, package BG560 , speed -4
Design statistics:
Minimum period: 21.564ns (Maximum frequency: 46.374MHz)
Maximum combinational path delay: 46.579ns
Maximum net delay: 19.611ns