Design Of Fast And Scalable Non-Blocking Scheduler For High Capacity Packet Router

Babu. M^{#1}, Krishnamoorthi. S^{#2}, Ravindran. G^{#3}, Rajkumar. S^{#4} Asst.Professor Department Of Ece^{#1} Mca^{#2,#4} Mba^{#3} Psrr College Of Engineering^{#1}, Psr Engineering College^{#2,#3,#4}, Sivakasi

ABSTRACT-As the need for non-blocking internet routers are becoming a need for high speed networks, I have designed an architecture based on high speed networks with the use of a new design of sequential greedy scheduling algorithm (SGS), is a maximal matching algorithm that provides non-blocking in an internet packet routers, when the traffic is policed. In this architecture, packets are split into cells of fixed length and stored at the input ports. The overall function of the system is implemented using Modelsim and output characteristics of the design are done to finalize the proposed design. This project increases the speed of packet transmission.

Key Words- High capacity, non-blocking, packet switching.

1. INTRODUCTION

The fast growth of bandwidth demand on the Internet has led to a need for high capacity packet routers, with large number of ports and high port speeds. Routers with input buffers are the most scalable single-stage routers. SGS algorithm speeds up the transmission of packets without any delay.

2. DESIGN OF THE SCHEDULER

The internal architecture of the input port is given in Figure1. The input port consists of several components: network processor, data memory, linked list memory, queue manager, and output selector. When a packet arrives to the router, its destination IP address is read by network processor (NP) and the router output to which the packet should be sent is determined. The NP also divides packets into smaller fixed length cells and stores them in the appropriate virtual output queue (VOQ).



Fig.1 Internal architecture of an input port i

In each input buffer, there are N VOQs that comprise cells bound for particular outputs. Data memory stores the incoming packets/cells until they are scheduled and sent through the switching fabric. Linked list memory stores the data memory addresses of cells in VOQs. Queue manager performs operations on virtual queues. Output selector calculates the schedule for the outstanding cells and stores it in output memory until the cells are read.

2.1. Linked List Memory

Cells in data memory that are bound for the same destination form a VOQ. There are N VOQs, corresponding to N outputs. Linked list memory stores address of cells in different VOQs and addresses of empty locations.

Each location in one virtual queue linked list (VQL) contains the address of the next location in that VQL, or zero (NULL) if no more locations belong to the VQL. Similarly, each location in the empty queue linked list (EQL) contains the address of the next location in EQL. The size of the linked list memory is defined by the number of cells that ought to be stored in the data memory of the router. The data memory should be able to store the number of cells equal to the frame length F. Thus, the linked list memory has F locations.

2.2. Queue Manager

Queue manager performs operations when a cell arrives to the queue manager, when it is scheduled by the output selector, or when it departs the router. The queue manager stores pointers to VOLs. There are three pointers to each VOL: to the beginning of the VQL, the first unscheduled packet in the VQL, and the end of the VQL. Similarly, the EQL is managed with two pointers: to the beginning and the end of the EQL. The pointers are updated each time one of the operations is performed. Multiple operations are carried out in the same time slot, and in general on different VQLs. At the beginning, all memory locations belong to EQL, and each location contains the address of the next location in the memory (except the last that points to NULL). The memory location is removed from the beginning of the EOL to the end of the VOL of some output, when a cell bound for that output arrives to

the router. The cell is stored to the address from this memory location.

When input sends a cell to some output, the cell address in the data memory is read from the heading location of the corresponding VQL, and this memory location is added to the end of EQL. When a cell is scheduled, the corresponding VQL is updated. If the pointer to the first unscheduled cell from the same VQL points to the last cell in that VQL, the pointer is set to NULL. Else, the pointer to the first unscheduled cell is to the next element in that VQL. The queue manager was designed in VHDL, and it can be scaled easily for different router sizes.

2.3. Output Selector

The output selector of the associated input schedules a cell by choosing the first available output from the set of outputs for which the given input has cells to send.





The result of the scheduling process is a vector in which only one bit, which corresponds to the scheduled queue, is set to logical one. The structure of the optimized output selector is shown in Figure2. Bits denoted by R contain information about cells of a particular input, which participate in the contention process. Bit R_j is set to "1" only if there are unscheduled cells for the j th output port, and the j th output port was not selected by previous input ports. As a result of the scheduling process, Q_j is set to "1" when the j th output port is chosen by the given input.

The information about the remaining output ports is forwarded to the next input port in the chain. The output selector is implemented recursively. The output selector for a router with 2^{K+1} output ports can be built recursively by using two output selectors for 2^{K} ports and one two-port output selector.

3. PACKET SWITCHING

The technique of packet switching was designed especially for the efficient transmission of computer traffic. When using packet switching, all data transmitted by the network user are divided into relatively small fragments known as packets. Each packet is provided with a header containing an address, which is necessary to deliver the packet to the destination. The presence of address in each packet represents one of the most fundamental properties of the packet-switching technique, since each packet may be processed by the switch independently of other packets of the information flow. Besides the header, the packet has another auxiliary field, which is usually located at the end of the packet and therefore is generally known as trailer. The trailer contains the checksum, which allows you to check if the information was corrupted during transmission. Packets are supplied to the network without previously reserving communication links, and at the rate at which source generates them. This rate cannot exceed the bandwidth of the access link.

4. SIMULATION RESULTS

SIMULATION RESULT OF DATA MEMORY

K	- default												
ik Edit	t View Inset Fo	rnet Tools Window											
30	a 140	ANH	Ī	()	1000]						- <i>1</i> 1 - N	
	kan/dk kan/adies		0.)01	(10) (10)	111	11		18	111	100	101	(10)	10
	han/dalari han/dalarut han/ha	101010	1007) 11.]0001	<u>)</u> (11)		, initia	<u>)</u> 0100	k per pere	1744444)110)""0	0110	,1011Q1
1													
1			0	.00.100	0***10*1*	10 11100-11) 	1010111111	0101010101	inmi:	101010		

SIMULATION RESULT OF OUTPUT SELECTOR

CUL HOF JOST	Fornet Tools H	rdw			
88 19	64 <u>}</u>	H		1000	
i Apale	ĺ				
/apoel/i	101	100		101	jii jii
/opes//2	0011	0101	,110)111	M1
l ipealit)IIIII)1001
i Aquelia					
lipelidi					
l ipielidi					
had 12					
(ipeld)					
Appelled					
l Josel lei					
/melei					
i indii					
and 2					

QUEUE MANAGER SIMULATION RESULT

HENE	e - default																									
e Edit	t Vew Insert For	net Tools Window																								
	5 13B.	H D X	(IT	G.L.		P-24	3	ยอง	13																	
	0.06	asc	111	30	149	Br A	Tel O	a 9 :		_	_	_	_	_		_			_			_			_	
1				Π.		л.		Π.						Π_				-1-		-	Π.	Π.	Π.		Π.	
	kpeenti	1																4								
1	/queuen/talan	1011100	101110			-			100000	<u> </u>			TOTAL	-					00011	<u> </u>		-				
-	And the second s		uu.																		20110	<u>u</u>				Ð
	And the second s																									
	heephth	lim	m	=		=		=		=	=	=	-				-	+	-	-	1001	1000	Wei	Ynti	Ynn	T
ł.	/queuen/wph		m		3001	100	301	00	0101	0110	0111	1000	101	100	1011	1100	0101	0110	0111	000						ľ
-			1100		10001	20010	10011	20110	CONTR	0011	100111	01000	11001	1:110	0011	0110	Statt	011	(1111	1000	10111	20110	10001	0100	00011	10
E.																		1								
-			m		1001	1010	301	m	(RIG)	0110	0111	100	htt	100	111	1100	1101	1110	0111	m	m	100	m	m	001	0
1																				_						
1								-										-				-	_			
2																										
H	Aqueuentine																									
	ingenerated and the second sec			h										-	-	-		+								
	Annerit																									
	Annen listen	10111980	100110	=	-	=	=	-	-	-	10.011	=	-	-	=		-	÷		-	-	⊨	-	-	=	+
1	laeum/deaut	0000000	UILL	10							102,111								0000111	-	10110	0	-			'n
					-																					T
5																										
-		000	m															t			m	in)	m	jnu	0101	C
1			1111		1001	1010	1011	(010)	(lili)	1111	10111	100	101	111	1011	110	(1101	1110	(1111	100						
-					1007	10010	00011	m	00101	<u> ()</u>	<u>)</u> 0111)700	10101	11110	001	<u>un</u>	(111)	01	(iIII)	1000	10001	101110)inii	010	(0011	0
1																										
1			m		2001	0010	001	00	010	0110	<u>)</u>)111	100	10	100	1011	1100	0101	110	1111	000	1001	100	101	000	(00)	0
5					-													-								
	rqæven enpy		-																							
	la nomba		-																							
H	lanumine Janumine		-																							
	heeven		-	-						-								+					=	-	-	÷
F	Appeuro Noran Ick									n		h			h.			h								
ā.			m		1001	2010	3011	2010	10101	13110	10111	1000	bor	1.70	1111	1100	0101	0110	0111	300	1001	100	1011	000	001	1
-		10111020	101110						(000110		100111		1011100	d .					.00011	0						ï
-		uuuuu	uu.	ίω																	101110	0				(
1																										
1																										
E	/queuen/Boran/te	1																								
				40111		300111.	800			110111	A1011.	a011.	3011.	9011	10011	0011	1000	.0 11	0011	8011	0010111	.eu 1011	001111		10000	113
				inc. 19	hurre			Tunnes	1	i nore	10111-1		1000												itmu=	_
	Nov	3900 m			26		43		600	-	80		la la	1	30	1	430		1603		300		2œ		206	
_	Curo 1	1527 m																1527=								
		1 >																								

SIMULATION RESULT OF LINKED LIST MEMORY

wave - default

File Edit View Insert Format Tools Window

REA THEN TARA LE COCCEN DIDIE

i filitik i filitik		1			٦	٦		1	٦	Π			٦	Π.	٦.	٦	٦	٦	7		7	
⊒ _ /hk/talan	11101110	11	. (1101	10				F									E					
B <mark>-1</mark> Additional <mark>1</mark> Additional	1190110 1						11101	10		<u>01101.</u>))1100.)"W	1	111001	1)000111	1	0001	0	100011	0	
, Anklidal ⊒-1, Anklida							1110	1101	110	1011	3010)III)W	<u>(</u> 111	0110	100	<u>)</u> 10	01	(01))001	, 000	(111
B− /ni/npt B− /ni/bjech			o proc I proc)111) ()1110	(1111 (01111))))]))))		0110	0110	01100	101	1000	m	11000	100111	10110	<u>)</u>	<u>)</u> []]	10011) 0010	<u>,</u> 0001	, 000
Animen B-M Aniadd Animul		11	0 (1101)110 	(1111		1110	1101	1100	1011	1 010	<u>)</u> 01	300	<u>(</u> 111)0110 	<u>)</u> 001) 0100	<u>1011</u>	<u>,</u> 011	<u>)</u> 001	(000	(1111
fiklend) fikles fikles																						
i Tekhe Tekhe																						

SIMULATION RESULT OF NETWORK PROCESSOR

i kave - del	www.cdaut												
File Edit View	i beet Fornat	Tods Window											
688	1984	ANES		9 1003									
₽ _ / hpipa	olet 11	nin illinin <mark>i</mark>				1101011110101010000							
B ri kpd		11101 [0110)))))	00			jimi				
₽ ¶ <i>h</i> ød		1010 0	DITI		h-))	10							
B ≓ kpû		1000 - (10011		[11]	01							
🕂 kpići		8111 (11100			W			<u>100000</u>				
🕂 kpiši	l I	inin <mark>(</mark>	111010		101	10							
🛃 / kp/di		IIII (i i i i i i i i i i i i i i i i i i i	11)							
₽ , kpi≦		1010 <mark>(</mark>	000			10			0000				
₽ , hplð		11000 🛛	10011		(III)	01)111000				
🔐 hajdi		in i	101000			0			1. 10 10 10 10 10 10 10 10 10 10 10 10 10				
₽ ¶ høids		1010 <mark>(</mark>	1110010)011	10)00100				
₽ <i>l</i> pi		11111 (00101		jm	11)							
3. hold		1810 <mark>(</mark>	0000			10							
₽ , ipid		1000 🛛	10011			01);;;;;()()				
₽, /pď		1111 (101000))))	0			100111				
₽ , lipii	4 🔟	icitio (j	1(****)		in in	10							
₽ , kpi		11111	10001		JU	11)							

SIMULATION RESULT OF SCHEDULER



[4]N.Mckeown, M, lzzard, a, Mekkittikul, W, Ellersick, and M. Horowitz, "The Tiny tera: A packet router core," IEEE Micro, vol,17, no,1, pp,jan-feb1997.

[5] A.Smiljanic, "Bandwidth reservations by maximal matching algorithm" IEEE commun.Lett,vol.8,no,,3 pp,177-179,Mar,2004.

[6]M.Petrovic, M,Blagojevic, "Design, implementation, and testing of the Controller for the terabit packet routers" in proc,IEEE ICCCAS,2006,vol,3,pp1701-1705.

5. CONCLUSION

Proposed design of the scheduler for the nonblocking internet router based on the SGS algorithm optimized the scheduler sub components so as to provide lower packet delays. The proposed scheduler can be used in high capacity packet routers that provide delay guarantees.

REFERENCES

[1]A, Smiljanic, Milos petrovic, Milos blagojevic, "Design of the Switching Controller for the Highcapacity Non-blocking Internet Router."IEEE Trans.vlsi system,vol,17,No,8,August 2009.

[2]A,Smiljanic, "Flexible bandwidth allocation in terabit packet routres", in proc, IEEE HPSR, Jun 2000,pp,233-241.

[3]H.J Chao, "Saturn: A terabit packet router using dual-round robin" IEEE Commun, mag,, vol,38,no 12,pp,78-84,dec 2000.