International Journal of Future Computer and Communication, Vol. 1, No. 4, December 2012

Efficient Bandwidth Allocation for Packet Scheduling

D. Rosy Salomi Victoria, and S. Senthil Kumar

***Abstract*—Quality of Service provisioning, reduction of system delay, service fairness and effective resource utilization are all critical issues in the job scheduling design. Our scheduler gives a chance to all classes to have access to the bandwidth. This model can achieve load balance among resources of the same class and simultaneously decrease queueing delays for waiting packets. In order to avoid serious performance degradation for packets of lower classes, one resource handler is set up for all classes; not only to deal with packet dispatching but also to migrate packets between classes when necessary. User favors application service with a higher percentage of bandwidth in order to provide QoS. There is no packet drop in our implementation. This scheduler performance has been shown by comparing with Output Controlled Round Robin algorithm in terms of transmission rate, jitter, bandwidth, scheduling and latency.**

***Index Terms*—Bandwidth based round robin, guaranteed service, maxmin fair scheduling, robust opportunistic scheduling.**

I. Introduction

Differentiated Service (DiffServ) [2] architecture has Internet Protocol flows classified and aggregated into different classes, marked with different priorities at the edge routers and dropped at the core routers. In DiffServ[3] networks, a customer makes a contract with the service provider for the establishment of a service profile. Due to the difficulty in predicting the bit rate requirement of VBR video sources, video channels may utilize more than the reserved bandwidth. The reduction of queueing delays can improve system performance and increase the degree of user's satisfaction. The QoS machines have a higher power of computation and high priority packets are only dispatched to this kind of machines. This machine can accept low-priority packets. The normal machines have lower computational capability and only execute low-priority packets.

II. Review of Existing Scheduling Schemes

The schedulers can be classified as work-conserving or non-work-conserving. A work-conserving scheduler is never idle if there is a packet awaiting transmission. Examples include Generalized Processor Sharing (GPS), packet-by-packet GPS also known as Weighted Fair Queuing (WFQ), Virtual Clock (VC), Weighted Round-Robin (WRR),Self-Clocked Fair Queuing (SCFQ), and Deficit Round-Robin (DRR). A non-work-conserving scheduler may be idle even if there is a backlogged packet in the system because it may be expecting another higher-priority packet to arrive. Examples are Hierarchical Round-Robin (HRR), Stop-and-Go Queuing (SGQ), and Jitter-Earliest-Due-Date (Jitter-EDD). Static scheduling may not be able to immediately meet user's requirements for different QoS. Some applications need a high-quality service to receive a short response time, but some others can accept a lower level service due to limited budgets. Per Hop Behavior are implemented at DiffServ network using scheduling mechanisms: Dynamic Deficit Round Robin (DDRR), Hierarchical Dynamic Deficit Round Robin (HDDRR) [7], Surplus RR (SRR), PQWRR (Priority Queuing WRR) [4], DRR+ (Latency Critical [LC] flows), DRR++ (Bursty LC flows), Core-Stateless Fair Queueing (CSFQ) [1], Rainbow Fair Queuing (RFQ), Dynamic Batch Co-Scheduling (DBCS [1]), Distributed Cooperative and Opportunistic Scheduling (COS)[1] , Smallest Message First (SMF) , Shortest Hop-Length First (SHLF) scheduling, OCRR (Output Controlled RR)[6] and Output Control Grant Round Robin (OCGRR) [5]. OCGRR supports all the three classes of traffic, maintained fair bandwidth allocation for competing network streams and reduced inter transmission time of packets from the same stream.

Manuscript received March 13, 2012; revised June 15, 2012.
D. Rosy Salomi Victoria is with the Dept. of CSE, St. Joseph's College of Engineering, Chennai, Tamilnadu, India (e-mail: rosysalomi@yahoo.com)
S. Senthil Kumar is with the Dept. of EEE, Government College of Engineering, Salem, Tamilnadu, India (e-mail: sengce2009@yahoo.in)

III. Improved Scheduling Technique

The proposed system architecture is shown in Fig. 1. The simulation shows how the packets are sent and reply is received by the destinations.

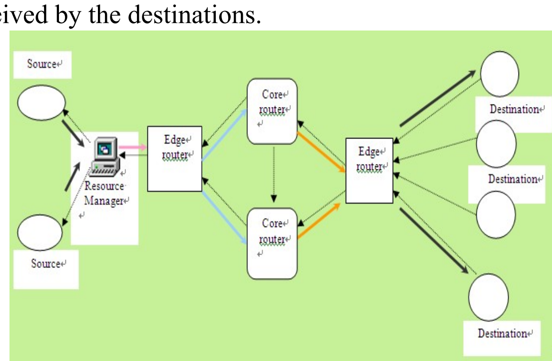

Fig. 1. Bandwidth based output controlled round robin technique

A. Bandwidth Based Output Controlled Round Robin Scheduling Algorithm

Scheduling happens in the core router. Scheduler will assign a bandwidth based on the packet sizes. The bandwidth is spitted into four categories as follows: higher bandwidth, medium bandwidth, lower bandwidth and reserved bandwidth. We have utilized the bandwidth reusability concept. Initially, the reserved bandwidth is set as 0. While transferring the packets, if the allocated bandwidth is higher than the packet size .i.e., excess bandwidth, it will be added with the reserved bandwidth. Any priority data can utilize this bandwidth. If it is lesser than the packet size, then the packet will be split into two. The possible packet will be sent through the allocated bandwidth and the remaining packets will be sent through the reserved bandwidth.

B. Comparison of Existing System with the proposed Work

Let the expected frame length be 4400 bits at the start of the frame. Table I show a EF and BE classes buffer details. Here, Pi,k denotes k packets from stream i with X bits (e.g., P21=200) and Gi is the available grant of stream i. Table II shows the grant values, the output sequences and the total transmitted bits at the end of each EF round of OCGRR. The transmission of the EF packets is finished at the fourth round with the total transmission of 3,000 bits.

The transmit EF sequence: P11;P21;P31;P71;P12;P22;P72;P13;P73;P74. This sequence shows that packets from different streams have fair access to the bandwidth via a smooth schedule. We take the same EF and BE streams of the existing OCGRR for the proposed work. Table III shows how the packets are transmitted by adjusting the bandwidth in the proposed work. In order to transfer video file, the maximum bandwidth allocated is 1000 byte/sec. Similarly, for transferring audio file, it is 600 bytes, for text file, it is 100 byte and reserved bandwidth allocated is 200 byte/sec. In the Table III, PR: The packet size received in that particular round, B: The sufficient bandwidth used to send that particular packet, P: Priority, PT: The packet transmitted in that particular round, R: Reserved Bandwidth; Excess bandwidth of PR is added to R. If bandwidth allocated is insufficient in B, bandwidth is deducted from R. e.g. In round #4, EF3 has PR as 240 and R as 120. 240 bytes can be sent through medium bandwidth (i.e. 600 bytes).So 600 -240 = 360. This excess bandwidth is added to R i.e. 120 + 360 =480. So total R=480.

TABLE I: EXAMPLE STATUS OF STREAM

EF#1	P_{14} (140)	P_{13} (200)	P_{12} (300)	P_{11} (100)	G_1=500
EF#2	P_{24} (440)	P_{23} (40)	P_{22} (1100)	P_{21} (200)	G_2=1000
EF#3	P_{34} (240)	P_{33} (740)	P_{32} (400)	P_{31} (500)	G_3=200
EF#7	P_{74} (200)	P_{73} (100)	P_{72} (200)	P_{71} (100)	G_7=450
BE	P_{B4} (350)	P_{B3} (200)	P_{B2} (300)	P_{B1} (400)	G_{BE}=1200

TABLE II: PACKET TRANSMISSION ORDER FIR OCGRRR

Round #	G_1	G_2	G_3	G_7	Output sequence in OCGRR	Total bits
1	400	800	-300	350	P_{11}, P_{21}, P_{31}, P_{71}	900
2	100	-300	-----	150	P_{12}, P_{22}, P_{72}	2500
3	-100	-----	-----	50	P_{13}, P_{73}	2800
4	-----	-----	-----	-150	P_{74}	3000
BE	G_{BE}=-50				P_{B1}, P_{B2}, P_{B3}, P_{B4}	4250

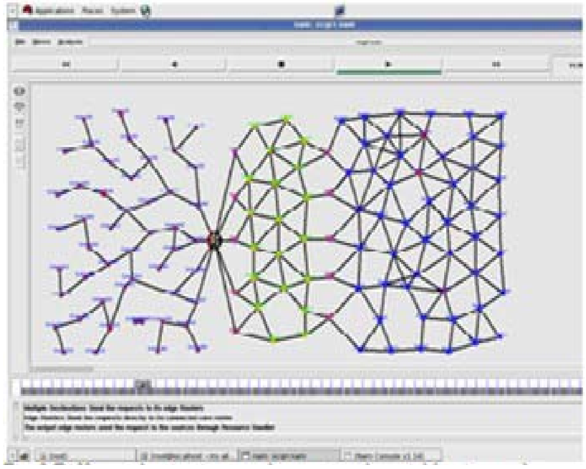

Fig. 2. Different destination sendrequest to the neighboring node and reechoes edge router

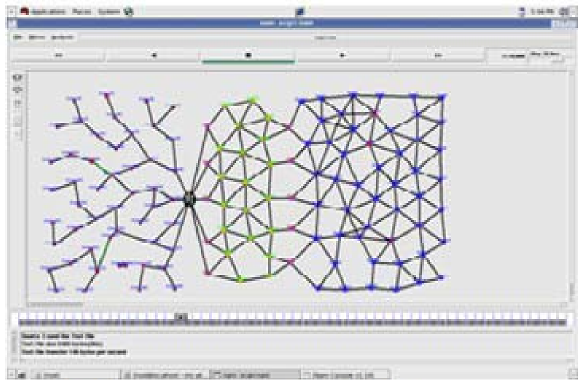

Fig. 3. core router scheduling replied data

TABLE III: PACKET TRANSMISSION ORDER FOR BOCRR

		Round #1			Round #2		
		P	*PT*	***R***	*P*	*PT*	*R*
EF1	*PR*		100	200		300	400
	B	L	100	200	L	100	200
EF2	*PR*		200	200		1100	200
	B	L	100	100	H	1000	300
EF3	*PR*		500	100		400	
	B	M	500	200	M	400	500
EF7	*PR*		100	200		200	500
	B	L	100	200	L	100	400
BE	*PR*		400	200		300	400
	B	M	400	400	L	100	200

	Round #3			Round #4		
	P	*PT*	*R*	*P*	*PT*	*R*
EF1		200	200		140	420
	L	100	100	L	100	460
EF2		40	100		440	460
	L	40	160	L	100	120
EF3		740	160		240	120
	M	600	20	M	240	480
EF7		100	20		200	480
	L	100	20	L	100	380
BE		200	20		350	380
	M	200	420	L	100	130

We have used NS2 for the proposed Bandwidth based Output Controlled Round Robin Scheduling. We have designed to use 50 destinations to send request, 50 sources to send reply, 10 edge routers to forward the data and 20 core routers for scheduling. We used this much count in order to do performance evaluation by comparing with the existing systems more in future. Multiple destinations send the request to its input edge router as shown in the Fig. 2. The edge router forwards the request directly to its connected core router. If the core router is directly connected to the output edge router then handover the request. If not, it handover the packets to other core routers which have been interconnected to it and reach the output edge router. The output edge router sends the request to the requested source through the resource handler Resource handler checks the multiple requests and forwards the request to the appropriate sources by passing through the neighbor nodes. The individual sources send the reply back to the resource handler. The resource handler gets the immediate reply and forward it to the nearby edge router. This router fixes the priority for the incoming data. Text files are assigned a lowest priority, audio files as the medium priority and video and image files as the high priority. Edge router sends the reply data to its core router. If this core router is busy by doing scheduling for other reply, then the packets will be forwarded to the interconnected core router which is available i.e., free. Both the core routers will simultaneously schedules their data's as given in Fig.3. After scheduling, the core routers forward the packet to the nearby edge router in the destination side. Then the edge router sends the data to the destination node by passing through the neighbor nodes. Thus the data packets are transmitted in an efficient order.

We set the simulator objects, create the nam and trace file, set the distance variables, define different colors for nam data flows, define nodes configuration, create the wired nodes, set the initial positions of nodes for destinations, edge routers, core routers, sources and resource handler, setting animation rate as 15ms, set the node sizes, set colors for all the nodes, finding the neighbor source, destination and core router nodes, print the distance between adjacent nodes, print the number of neighbor node for each node, open files to write data in order to draw graph, assign full duplex link, given bandwidth, delay and queue type.

We initialize check as 500 and print it, calculate check = [expr round([$size_ value])], if check < 2000 then set destination nodes 44,31,7 as red color, set source nodes 14,3,33 as red color, print as "Multiple Destinations Send the requests to its edge Routers", create a second TCP sink agent and attach it to another node, connect tcp2 source to tcp sink at neighbor node, create a second FTP source "application" for the following: print "Resource Handler checks the multiple requests and forwards the requests to the appropriate sources, Individual source sends their reply data to the Resource Handler, The Resource Handler forwards it to the next input edge routers, Source 14 sends the Video File, Video File size 62400 bytes, Video File transfer 1040 bytes per second, Input Edge Routers send the reply to its core routers", connect resource handler n edge router thru TCP, print" Scheduling Process, Higher Bandwidth 1000 bytes, Medium Bandwidth 600 bytes, Lower Bandwidth 100 bytes, Reserved Bandwidth 200 bytes", connect edge router n core router thru TCP, print" bandwidth 1000 bytes packet size 1040 ,1000 bytes allocated and 40 bytes sent in Reserved, total reserved bandwidth 240", print" Source 33 send the Audio File, Audio File size 32400 bytes, Audio File transfer 540 bytes per second", print "bandwidth 600 bytes packet size 540 ,540 bytes allocated and 60 bandwidth is added in reserved, total reserved bandwidth 300 bytes", print "Source 3 send the Text File, Text File size 8400 bytes, Text File transfer 140 bytes per second, bandwidth 100 bytes packet size 140 ,100 bytes allocated and 40 bytes sent in reserved, total reserved bandwidth 340 bytes", print the first 5 arrival times and sizes. Print the graphs in gnuplot. Finally, execute nam file.

IV. Performance Evaluations

The Bandwidth based Output Controlled Round Robin (BOCRR) is compared with Output Controlled Grant Round Robin (OCGRR) [5] in terms of transmission rate, jitter, bandwidth, scheduling and latency. This ensures that the BOCRR is better than OCGRR and it's evident from the graphs. In the graph given in the fig. 4 the transmission rate is high in BOCRR. BOCRR have less average start-up latency in the graph shown in the Fig. 5. In the graph given in the fig. 6, the jitter is low in BOCRR.

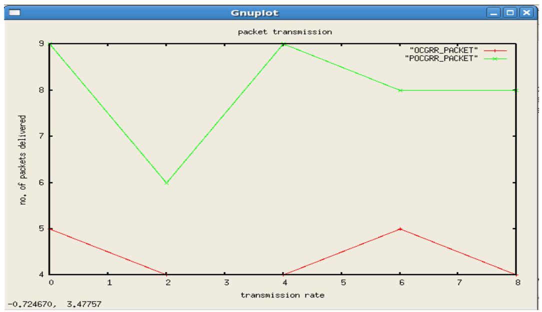

Fig. 4. Comparison of BOCRR with OCGRR in terms of transmission rate and no. of packets delivered

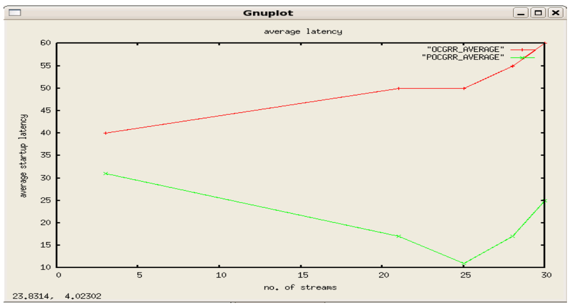

Fig. 5. Comparison of BOCRR with OCGRR in terms of latency and the no. of streams

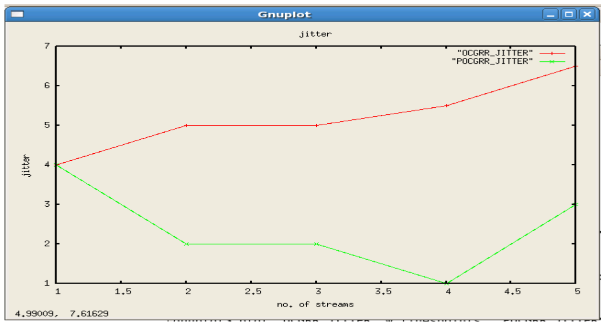

Fig. 6. Comparison of BOCRR with OCGRR in terms of jitterand the no. of streams

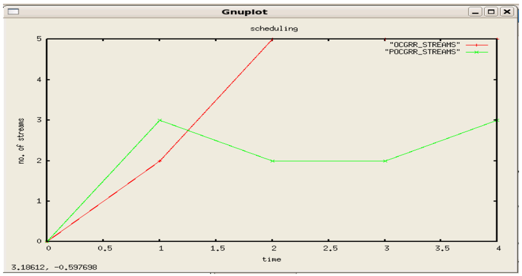

Fig. 7. Comparison of BOCRR with OCGRR in terms of scheduling time and the no. of stream

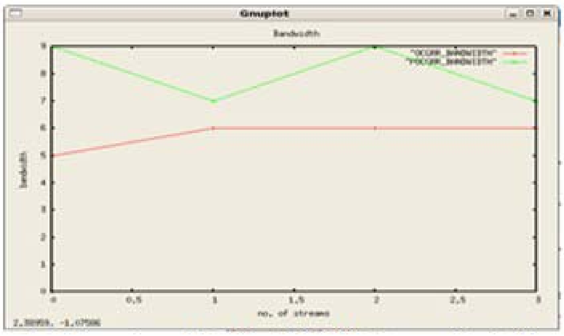

Fig. 8. Comparison of BOCRR with OCGRR in terms of bandwidth and the no. steams

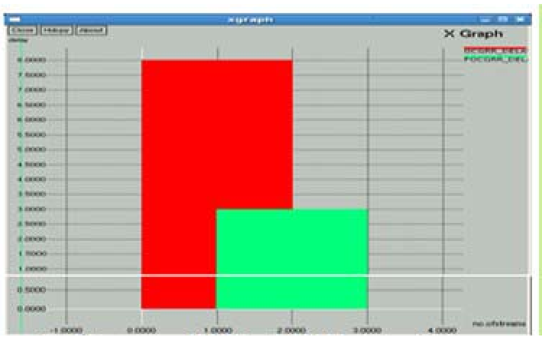

Fig. 9. Comparison of BOCRR with OCGRR in terms of delay and the no. steams

BOCRR took lesser time to schedule the packets in the graph shown in the Fig.7. In the graph given in the fig. 8, the bandwidth utilized is more in BOCRR. We are planning it to be reduced by using some new scheme. The comparison of OCGRR with BOCRR in terms of the number of streams and the delay is shown in the fig.9. BOCRR delay is much lesser than OCGRR.

V. Conclusion

Our scheduling algorithm has the abilities of reducing the stream inter transmission time. The starvation of lower priority class has been avoided. The streams fairness has been ensured. It provided service level agreement by guaranteed output. There is no packet drop. The performance of this Bandwidth based output controlled round robin scheduler has been compared with the existing OCGRR algorithm in terms of transmission rate, jitter, bandwidth, scheduling and latency. The graphs ensured that the BOCRR is better than OCGRR. Some more performance analysis will be done in future by comparing the new technique with other existing scheduling schemes.

References

[1] C. Annadurai, "Review of Packet Scheduling Algorithms in Mobile Ad Hoc Networks," in *Proc. International Journal of Computer Applications*, vol. 15, no. 1, pp.7-10, 2011.

[2] M. Li and B. Song, "Design and Implementation of a New Queue Scheduling Scheme in DiffServ Networks," in *Proc. IEEE*, pp.117-122, 2010.

[3] J. Musacchio and S. Wu, "The price of Anarchy in Competing Differentiated Services Networks," in *Proc. IEEE Technology and Information Management Program*, WeD3.1,pp. 615 -622, 2008.

[4] Y. Zhang, "Performance of an Integrated Scheduling of Priority Weighted Round Robin and Strict Priority in DiffServ Networks," vol. 6, pp.1-13, 2007.

[5] A. G. P. Rahbar and O. Yang, "OCGRR: A new scheduling algorithm for differentiated services networks," in *Proc. IEEE Transactions on parallel and distributed systems*, vol. 18, no.5, pp. 697-710, 2007.

[6] A. G. P. Rahbar and O. Yang, "The Output-Controlled Round Robin Scheduling in Differentiated Services Edge Switches," in *Proc. IEEE BROADNETS '05*, pp. 237-245, Oct. 2005.

[7] C. C. Wu, C. Moh, H. M. Wu, D. J. Tsaur, and W. Lin, "Efficient and Fair Hierarchical Packet Scheduling using Dynamic Deficit Round Robin," in *Proc. Communications and Computer Networks*, 2005.