

Design and Implementation of Pipelined DRR ASIC

Yi-Mao Hsiao, Ming-Jen Chen, Yier Chen, Yuan-Sun Chu and Cheng-Shong Wu

SoC Laboratory, Department of Electrical Engineering ,
National Chung Cheng University,
Chia-Yi 621 Taiwan, R.O.C
93mowmow@vlsi.ee.ccu.edu.tw

Abstract—In this paper, a novel scheme called Pipelined Deficit Round Robin (PDRR) is proposed for packet scheduler. We reorder the processing stages and use the pipelined method to reduce the delay time. The scheme preserves $O(1)$ complexity in hardware. By the simulation result, PDRR has 60% improvement better delay time than DRR. The ASIC operates approximately 3.5ns by 0.18 um CMOS technology and supplied with 1.8V and power dissipation is 47 mW. The area is 0.845 mm X 0.879 mm involving pads. It can furnish approximately 285.71 MHz so that is enough to satisfy OC-768.

Keywords-pipelined DRR; Quality of Service; scheduling; ASIC

I. INTRODUCTION

Recently numbers of internet applications are growing rapidly, such as Voice on IP (VoIP), Video on Demand, Internet Gaming and Internet Conference. These services are different from ordinary data transferring that the internet was designed to transfer data in the beginning. They need higher internet quality especially in delay time. To maintain internet's ability for providing enormous application demand, Quality of Service (QoS) is discussed[1][2][3][4].

One of the most essential components in QoS is packet scheduler. When there are numbers of queues being backlogged, scheduler decides which packets in the queues can be transferred. Scheduler has to divide the resource fairly. Therefore, fairness is one of the critical metric in designing a scheduler.

Deficit Round Robin (DRR)[5] is the most widely used scheduling algorithm. DRR provides high fairness and is simply to design. But the delay time of DRR is very serious. Some implementing methods such as Aliquem-DRR[6], WDRR[7], Pre-order DRR[8] and SRR[9] propose improvement and discussed advantage and disadvantage for DRR.

In this paper we propose a novel scheduling scheme called Pipelined Deficit Round Robin (PDRR). We implement the DRR with pre-calculation and pipelined design. PDRR achieve the same fairness as DRR with a shorter delay time than DRR, and is easily to implement by hardware. The ASIC of PDRR is enough to satisfy OC-768.

The rest of the paper are as follows. Section II introduces the Packet Schedulers. The PDRR scheme architecture and its implementation are described in section III. Section IV is the VLSI implementation. Section V shows our simulation results by using NS2[10]. And section VI is the conclusion.

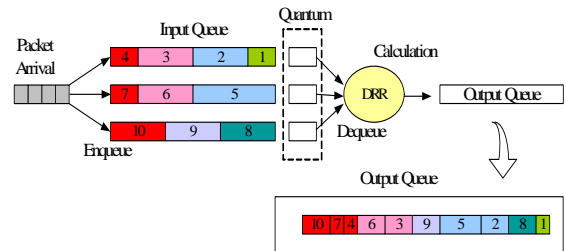


Figure 1. Architecture of DRR

II. PACKET SCHEDULERS

A. DRR Scheme

DRR scheme is one of the most popular scheduling algorithms nowadays. The advantages of DRR are simple and fairness. The architecture of DRR is depicted as Figure 1. There are many input queues for each flow. Each queue has a quantum to decide how many bytes can be sent in one round. The remaining deficit is added to the deficit of the next round. If the packet size is smaller than the quantum, it can be sent directly to the output queue.

According to the packet classification, packets are put in several input queues when they are pulled out from the input queues by DRR algorithm. Based on the experience of DRR scheme in Linux kernel 2.4, we figure out the three major processing steps of DRR shown in Figure 2. Enqueueing, Dequeueing, Calculation and the packets is put into the output queue.

DRR has some disadvantages in implementation. When a flow contains a large packet, the processing step repeats for several times from the queues. DRR spends longer time to process a large packet. Calculation has to finish tightly after dequeue. Therefore, each packet must wait for the previous packet to complete dequeue and calculation steps. Figure 2 shows that the time between enqueue and dequeue for each packet becomes longer and longer.

B. Aliquem-DRR Scheme

The architecture of Active List Queue Method DRR (Aliquem-DRR) is shown in Figure 3. Aliquem-DRR calculates each flow can be served, and stores the result in active lists. Aliquem-DRR indeed resolves the problem of DRR, but it requires an extra memory to store the active lists.

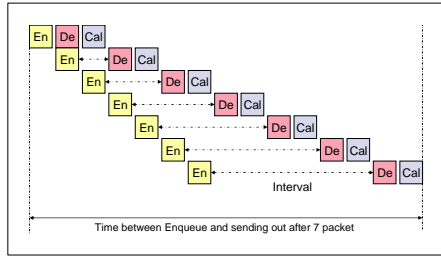


Figure 2. The processing steps of DRR

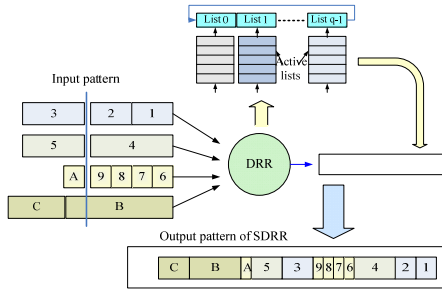


Figure 3. Architecture of Aliquem-DRR.

C. SRR Scheme

Smoothed Round Robin (SRR) improves the delay and burstiness properties by spreading the data of a flow to be transmitted in a round over the entire round using a weight spread sequence. SRR smoothes the round-robin output burstiness by allowing a flow to send one maximum length packet worth of bytes every time. SRR uses the Weight Spread Sequence and the Weight Matrix to schedule packets. However, it is not easily on hardware design.

D. Pre-Order DRR Scheme

Pre-order DRR scheme is proposed to improve the delay time of DRR. Input queue is classified in Pre-order DRR. As packets arrive, they are enqueued to different priority queue and then transmitted according to the rule of each priority queue. The Pre-order DRR scheme improves the delay time of DRR, but the order of sending packet is much different from the packet order of DRR. So, pre-order DRR implementation in hardware is also complicated.

III. PDRR SCHEME

Because of the disadvantages mentioned above, we propose a novel scheduling scheme to improve DRR algorithm called Pipelined DRR (PDRR). The system architecture demonstrates in Figure 4. We shift the calculation step in front of the enqueue step. The sending decision of packet during the present round is calculated before enqueued the packet. When there is a large packet in the flow, PDRR only pass one dequeue step however DRR has to do this step in several times.

We design the architecture with pipelined technology. Based on the same steps of DRR, the processing of PDRR scheme is divided in three steps: calculation, enqueue, and dequeue. Because we shift the calculation step forward, these

three steps become independent to each other. The system can process three steps at the same time for three different flows. As shown in Figure 5, using the pipelined architecture can improve the delay time.

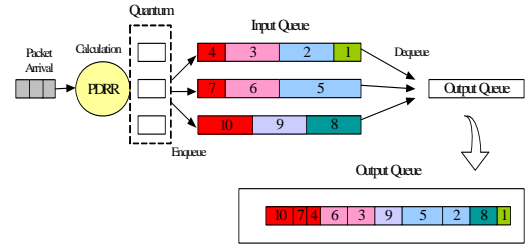


Figure 4. Architecture of PDRR.

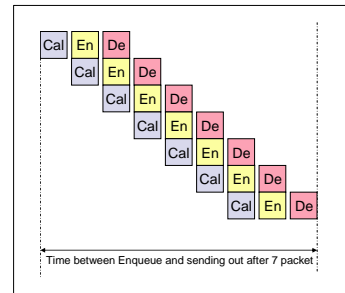


Figure 5. The time of pipelined PDRR.

The system updates the deficit when the packet arrived. Then it calculates how many rounds which the packet can be sent out by PDRR algorithm. We assume r as round number, q as quantum size, d as deficit size, p_i as packet size where $\forall_i \in$ number of packet and p_n as packet number. The Calculation algorithm of PDRR is shown in Figure 6.

```

Calculation_Algorithm(){
Input: packet size( $p_i$ )
Output: round count ( $r$ ) and flow deficit( $d$ )
For (any packet calculate the  $r$ ) {
    if( tmp  $d > p_i$ ){
         $d = \text{tmp } d - p_i$ ;
    }
    else if ( tmp  $d == p_i$ ){
         $d = q$ ;
        Return  $r$  and write to flow  $d$  ;
    }
    else{
         $d = \text{tmp } d + q$ ;
        Return  $r$  and write to flow  $q$  ;
    }
    tmp  $d = d$  ;
     $r = r + 1$ ;
}
}
    
```

Figure 6. Calculation algorithm of PDRR.

As Figure 7 shown, the blocks in the input queue represent the packets, the number inside the block is the packet number p_n . The number inside the bracket is the number of sending round r and which in the second row is the packet size p_i . The quantum size is 100 bytes for each flow. It shows the example of PDRR enqueueing process. Figure 8 shows the example of PDRR dequeuing process that the deq module output the packets by the round r ordering.

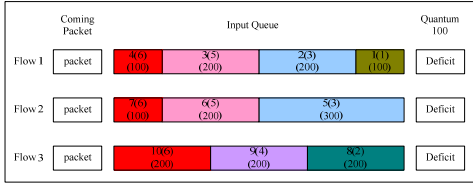


Figure 7. Example of PDRR enqueueing process.

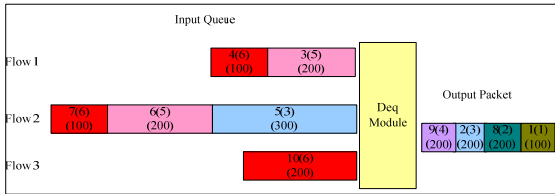


Figure 8. Example of PDRR dequeuing process.

IV. VLSI IMPLEMENTATION

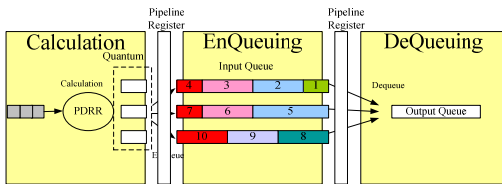


Figure 9. PDRR System Architecture.

As shown in Figure 9, we implemente the PDRR scheduling system with 3 stages pipeline. The major design of the system is the calculation module that carried out the PDRR algorithm. We use an adder/subtract to calculate the deficit of every flow and round counters to count the sending round of the packet. Figure 10 illustrates the architecture of the calculation module. The other essential design of PDRR is the dequeuing module. Traditional queue is implemented with FIFO. In order to reduce the dequeuing times, the FIFO was designed with an extra pin to read the data in FIFO without decreasing the read pointer. The dequeuing times can be decreased. There are six queues implemented with 24 bit width FIFO and depth is 16. We assume the system is Differentiated services networking. There are six queues: Best Effort(BE), Expedited Forwarding (EF), and Assured Forwarding (AF1~AF4).

The PDRR ASIC design is carried out in Verilog hardware description language and synthesized by Synopsys using CCU 0.18 um CMOS single-poly six-metal standard cell library. The circuit is about 107740 gates. Figure 11 shows the layout of PDRR ASIC. The area is 0.845 mm X 0.879 mm involving

pads. The simulation result shows that the chip operates approximately 3.5ns. The supply voltage is 1.8 V and the power dissipation is about 47 mW.

V. SIMULATION RESULT

We simulate the PDRR scheduling scheme by NS2[12]. We also simulate DRR, SRR and Aliquem-DRR then compare the PDRR with them. Figure 12 shows the simulation environment setting. The bandwidth of all links is 1M byte/sec. The translation delay is 10 ms. The link type is UDP. The packets are all constant bit rate (CBR) type. Total simulation time is 3.5 second shown in Table I.

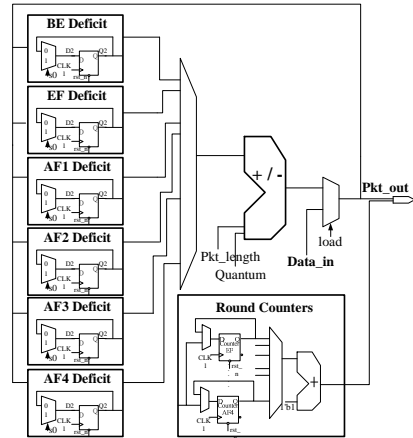


Figure 10. The Architecture of Calculation Module.

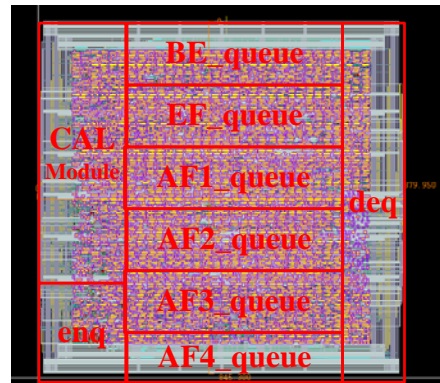


Figure 11. The Layout of PDRR Chip.

We change the packet size for each simulation. According to the analysis, the packet size and the number of packets affect the delay time. The larger packet usually needs more time to be enqueued and dequeued. Because of the pipelined architecture in PDRR scheduling scheme, the delay time can be decreased.

By the simulation result, the delay time of Aliquem-DRR and the PDRR is very similar and better than DRR and SRR as shown in Figure 13. But Aliquem- DRR needs more memory to store the active lists. We find that the delay time of PDRR is better than DRR about 60%, and better than SRR about 50%.

VI. CONCLUSIONS

In order to satisfy the requirement of many applications for the quality of network, many scheduling schemes have been proposed. DRR is one of the most popular schemes. It is easy to implement and has a high fairness, but the delay time of DRR is larger, especially at $O(N)$ work complexity. We proposed PDRR with reordering the processing stages and using the pipelined method to reduce the delay time. The PDRR scheme preserves at $O(1)$ complexity in hardware and has the same fairness as DRR. In our simulation, we make comparison between DRR, Aliquem-DRR, and SRR. As a result, we find that the delay of PDRR is better than DRR about 60%, and better than SRR about 50%. The delay time of PDRR and Aliquem-DRR is almost the same, but the PDRR is easily to implement on hardware. The PDRR ASIC design is carried out using Verilog hardware description language. It operates approximately 3.5ns by using a 0.18 um CMOS technology. The ASIC can furnish approximately 285.71 MHz so that is enough to satisfy OC-768.

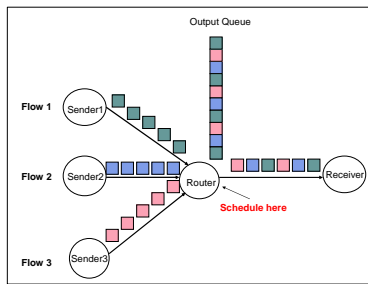


Figure 12. The Environment of Simulation.

TABLE I. THE SETTING OF OUR SIMULATION.

| | Sim.1 | Sim.2 | Sim.3 | Sim.4 |
|----------------------|-------|-------|-------|-------|
| Packet Size (Bytes) | 200 | 300 | 400 | 500 |
| Bandwidth (MB) | 0.4 | 0.6 | 0.8 | 1 |
| Total Bandwidth (MB) | 1.2 | 1.8 | 2.4 | 3 |

Time interval is 0.0005 sec. The setting of three flows are the same.

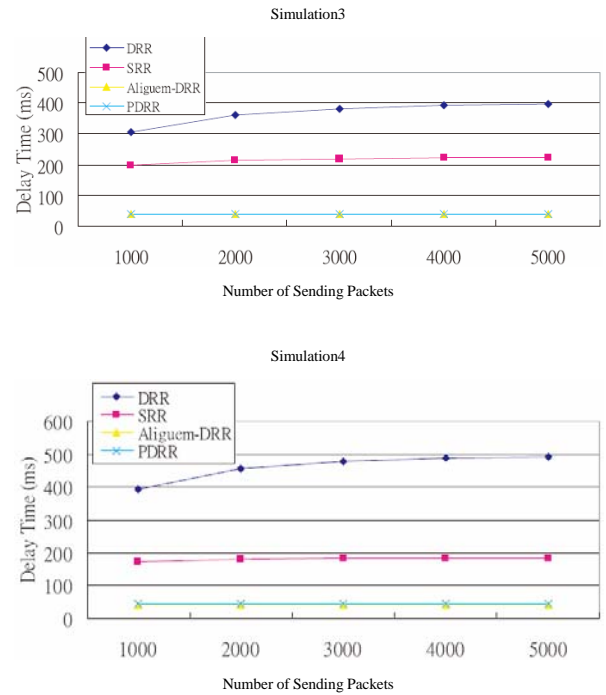
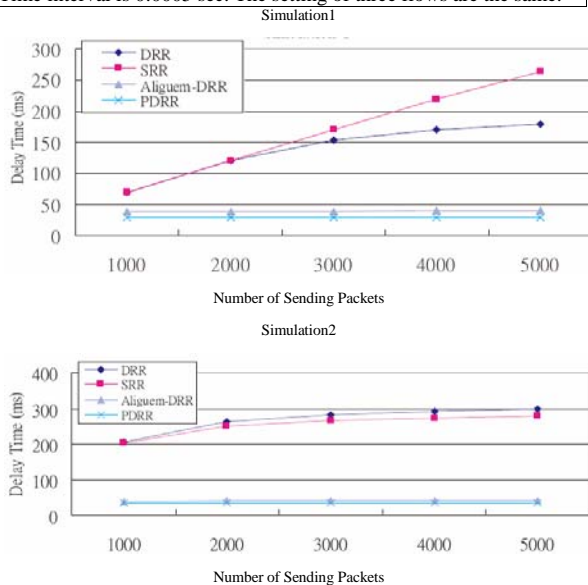


Figure 13. Simulation Results

REFERENCES

- [1] D. Clark, S. Shenker, and L. Zhang, "Supporting Realtime Applications in an Integrated Services Packet Network: Architecture and Mechanism," in *Proc. SIGCOMM'92*, 1992.
- [2] J. Kurose, "Open issues and challenges in providing quality of service guarantees in high-speed networks," *Comput. Commun. Rev.*, vol. 23, no. 1, pp. 6-15, Jan. 1993.
- [3] S. Floyd, and V. Jacobson, "Link-share and Resource Management Models for Packet Networks," *IEEE/ACM Trans. Networking*, vol. 3, Aug. 1995.
- [4] V. Fodor (née Elek), G. Karlsson, and R. Rönngren, "Admission Control Based on End-to-End Measurements," in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, March 26-30, 2000.
- [5] D. Stiliadis and A. Varma, "Efficient Fair Queuing Algorithms for Packetswitched Networks," *IEEE Transactions on Networking*, vol. 6, no. 2, pp. 175-185, April 1998.
- [6] L. Lenzini, E. Mingozzi, and G. Stea, "Aliquem: A novel DRR Implementation to Achieve Better Latency and Fairness at $O(1)$ Complexity," in *Proc. 10th Int. Workshop on Quality of Service (IWQoS)*, Miami Beach, FL, May 2002, pp. 77-86.
- [7] Chang Gui-Feng, "Dynamic Weighted Round Robin : A Simple Packet Scheduling Mechanism for Internet" , *Master Thesis*, National Chung Cheng University, Chia-Yi , Taiwan 1999.
- [8] S.-C. Tsao and Y.-D. Lin, "Pre-order Deficit Round-Robin: A new Scheduling Algorithm for Packet Switched Networks," *Computer Networks*, vol. 35, pp. 287-305, Feb. 2001.
- [9] G. Chuanxiong, "SRR: An $O(1)$ Time Complexity Packet Scheduler for Flows in Multi-Service Packet Networks," *IEEE Transactions on Networking*, vol. 12, no. 6, pp. 1144-1155, December 2004.
- [10] K: Fall and K. Varadhan, (Eds.) "ns Notes and Documents", The VINT Project. UC Berkeley, LBL, USC/ISI, and Xerox PARC, February 25, 2000. Available from <http://www.isi.edu/nsnam/ns/>.