A Deficit Round Robin Input Arbiter for NetFPGA

Jonathan Woodruff University of Cambridge
jonwoodruff@gmail.com

ABSTRACT

I have developed a lightweight Deficit Round Robin scheduler for the input arbiter of the NetFPGA router. The design is effective in reducing latency for small packets from one port in the presence of saturating traffic of large packets from another port. The design hides arbitration latency when multiple channels are active and achieves nearly optimal latency for the NetFPGA architecture. The design demonstrates that lightweight Fair Queuing algorithms can be useful for simple arbitration without introducing an additional queuing stage which increases both complexity and latency.

Categories and Subject Descriptors

H.4 [**Networking**]: Miscellaneous

1. INTRODUCTION

Fair Queuing requires some high-level knowledge of history and some high-level ability to rearrange packets which can be quite large. The requirement for high-level knowledge adds complexity and the requirement for arbitrary rearrangement of packets necessitates another queuing stage in the router which adds both complexity and latency. Deficit round robin scheduling, first formally described by Varghese in 1995 [4, 3], solves the knowledge problem by providing a simple mechanism for closely approximating a Fair Queuing algorithm with very little computational or storage overhead. However to properly reschedule packets a new queuing stage is necessary. While this complete and general approach has been applied for fairly managing arbitrary flows in powerful commercial routers, I believe the Deficit Round Robin selection algorithm can be very useful in the input arbiter of low complexity routers. If the input ports are the flows to be shared fairly, there is no need for another queuing stage as packets are already queued by input port in the input arbiter. This approach allows simple and inexpensive systems to leverage fair routing.

I have incorporated a Deficit Round Robin scheduler in the input arbiter of the NetFPGA router platform [2, 5] and demonstrate that it effectively allows small packets to flow through the router freely in the presence saturating traffic of large packets on another port.

2. PREVIOUS WORK

Network researchers have rigorously explored options for supporting Fair Queuing in network routers [1]. We will describe Fair Queuing, Deficit Round Robin, which is a practical approximation of Fair Queuing, and the implementation of Deficit Round Robin on the NetFPGA system.

2.1 Fair Queuing

Fair Queuing, proposed by John Nagle in 1985, is an algorithm that guarantees fair sharing of a router's bandwidth resources. Because packets of very different sizes can be sent through a router, simple round robin servicing of the input queues may result in large inequalities. For example, a file transfer may be sending very large packets and a voice-over-IP device may be sending small latency sensitive packets. If the large data packets are chosen with equal frequency to the small voice-over-IP packets the file transfer will use a disproportionately large part of the bandwidth. John Nagle proposed a theoretical model where the bits of each packet are serviced in a round robin fashion. This model would allow fair sharing of bandwidth resources but is impractical in that form because packets must be forwarded as complete units. Therefore John Nagle proposed a model to implement Fair Queuing where the scheduler computes the time that each packet would complete being forwarded in the theoretical model and sends the packet with the nearest time. This design is computationally expensive and other more practical but less precise methods for approximating Fair Queuing have been proposed.

2.2 Deficit Round Robin

Deficit Round Robin is one of the most successful algorithms to approximate Fair Queuing with minimal computational overhead [4, 3]. Deficit Round Robin places the size of each packet in a counter and decrements the counter by a quanta at each round through the queues and forwards the packet when the counter reaches zero. Thus a scheduler scheduling one queue with large packets and one queue with small packets will allow a proportional number of small packets through for each large packet forwarded. This simple algorithm emulates Fair Queuing with a very low computational overhead and is one of the most practical techniques for approximating Fair Queuing.

2.3 Deficit Round Robin on NetFPGA

The NetFPGA project at Stanford [2, 5] has a Deficit Round Robin module. This module is composed of two stages. The first categorizes packets into flows and the second reenqueues them by flows and forwards them according to the Deficit Round Robin algorithm.

As provided from Stanford, the Deficit Round Robin package for the NetFPGA can distinguish flows based on input port or the type of service tag. Input port is one of the most useful ways to divide flows for Fair Queuing as individual computers or networks attached to the router can be prevented from monopolizing the traffic through the router. However the packets are already queued by input port in the input arbiter of the NetFPGA and the extra steps of categorizing and reenqueuing them for Deficit Round Robin scheduling wastes resources and adds delay.

Furthermore the round robin scheduler in the input arbiter will feed packets into the system in round robin fashion from the input queues and packets will leave the system in the same proportion as they come in. Resorting them by input queue and forwarding them according to the Deficit Round Robin scheduling algorithm later in the pipeline will not actually change the bandwidth allocation behavior of the system. While the Type of Service categorization would be effective, the current Deficit Round Robin implementation for the NetFPGA is not able to affect the scheduling of flows categorized by input port. I have implemented a Deficit Round Robin scheduler directly in the input arbiter of the NetFPGA design to allow the input arbiter to approximate Fair Queuing on the four input ports with very little additional complexity or delay.

3. THE DEFICIT ROUND ROBIN INPUT ARBITER

My design is simple and adds minimal delay. I add counters to each of the four input queues and the logic necessary to extract the size of each packet and to decrement the counter by a register-set quanta. However to improve performance I have implemented two channels out of the input arbiter and into the output port lookup so that the next packet can be selected and begin lookup while the previous packet is being forwarded. A Deficit Round Robin based input arbiter follows others who have implemented Deficit Round Robin scheduling for simple queuing applications even in network switches [6].

3.1 Deficit Round Robin

Deficit Round Robin approximates Fair Queuing by attempting to forward an equal number of bytes from each flow. If queues are sorted into flows and the flows are contending, that is several flows have packets to forward, deficit round robin achieves a fair forwarding policy for the available packets. The size of the packet from the header is placed in a counter which is decremented at each round through the round-robin scheduler. If a counter reaches zero, it is forwarded. Thus ten 100 byte packets from one flow will be forwarded in the time of a single 1000 byte packet from another flow and fairness is achieved.

3.2 Counters

The FIFOs in the input arbiter of the NetFPGA design hold 4 elements of 64 bits each. The last frame latched into the input FIFOs contains the size field of the IP header. A state machine counts the frames entering the input FIFO and grabs the size field of the last frame and places it in a counter for that queue.

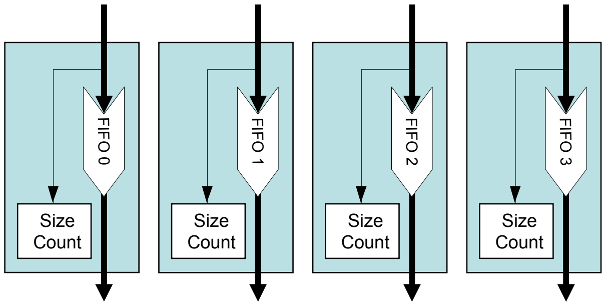


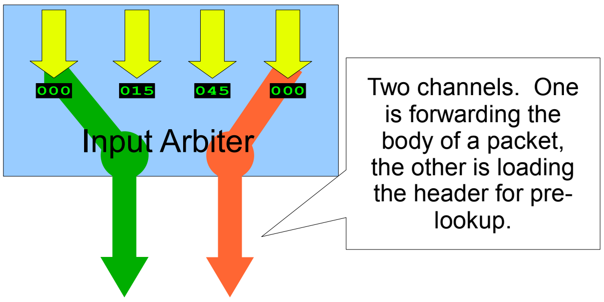
Figure 1: Deficit Round Robin Input Queues

3.3 Round Robin Scheduler

The round robin scheduler steps through the queues and decrements each counter by a quanta which is assigned as a register by software. If one counter reaches zero, that queue is selected and is forwarded to the next available output port. If the arbiter is under contention this arrangement forwards an equal number of bytes from each input port with queued packets.

3.4 Minimum Sizes and Pre-selection

Decrementing the counter upon arrival of each packet could add significant latency to each packet. We mitigate this effect firstly by subtracting a minimum packet size before loading the counter. Packets with sizes below the minimum size are forwarded immediately.



Two channels. One is forwarding the body of a packet, the other is loading the header for pre-lookup.

Figure 2: Dual Channels to Mask Selection Latency

Secondly, we may continue to decrement the rest of the pointers while a packet from one queue is being forwarded. If packets are present on more than one queue, the next packet may be selected before the current packet is finished being forwarded and zero added latency is observed. Unfortunately heavy traffic on a single port still sees the added latency. Additional logic could forward packets immediately if no other packets are waiting but this behavior would cause small infrequent packets to wait on average half of the forwarding time of the packets from the busy port. Adding latency between the packets of a busy port creates opportunity for small packets to be forwarded immediately if they are received during the idling time.

4. EVALUATION

We evaluate the Deficit Round Robin input arbiter my measuring its ping latency under contention. We flooded one port of the router with 1500 byte packets using a simple Unix utility and attempted to ping through another port on the router. The size of the ping packet is 56 bytes. A simple round robin system would delay the ping packet by half the time required to forward a 1500 byte packet on average but a system with Fair Scheduling would forward the ping packets immediately when they are received resulting in very low latency. We may attempt to tune our system to achieve the desired fairness behavior by changing the deficit quanta which is decremented from the counter on each queue. Figure 3 shows ping latency versus the size of the deficit quanta. We subtract a minimum packet size of 64 bytes so the counter of a 1500 byte packet begins at 1436. With a deficit quanta of one, a 1500 byte packet must wait 1436 cycles before being forwarded. Since the NetFPGA router forwards 8 bytes per cycle, the packet requires 188 cycles to be forwarded. Thus a system with a deficit quanta of one and a single saturated port achieves a utilization of only 11.5%. This effect is reflected in the total throughput of the system charted in figure 4 where maximum system throughput of 120 megabytes per second (or 960 megabits per second) is achieved with a quanta of 10 which achieves over a 67% duty cycle which is enough to saturate the output queues. A quanta of 1 gives a system throughput of 14 megabytes per second.

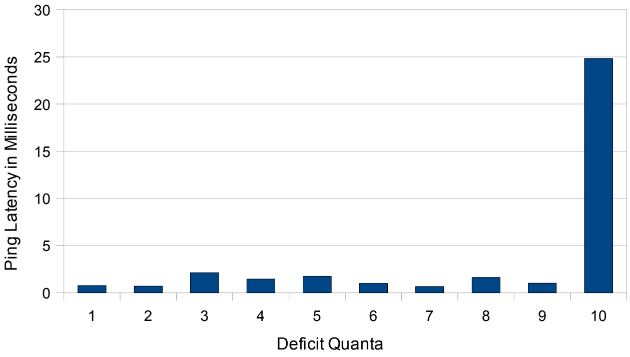


Figure 3: Ping Latency vs. Deficit Quanta

It is clear from comparing figure 3 and figure 4 that a saturated system has a very detrimental effect on ping latency. Latencies increase from less than 2 milliseconds in any case that is not saturated to 25 milliseconds in the saturated case. For this scenario a deficit quanta of 9 would be ideal to allow maximum throughput without saturating the system. For the case where all four ports may be loaded, a deficit quanta of 2 should ensure that the system is never saturated at the expense of limiting individual port bandwidth to 30 megabytes per second.

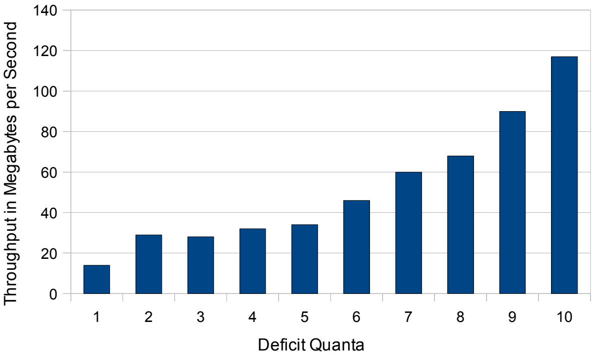


Figure 4: Total Throughput vs. Deficit Quanta

5. CONCLUSION

This Deficit Round Robin scheduler for the input arbiter of the NetFPGA demonstrates that Fair Scheduling properties can be achieved with very little added complexity to current designs. Deficit Round Robin input arbiters should be useful in simple routers and switches. They should be especially useful in the edges of a network where each port represents a single machine or a single cluster of machines that represent an independent sharing category. While Fair Queuing among more complex flows may be implemented at a higher level, Fair Queuing can still be applied with very low cost at the edges of the network.

6. REFERENCES

[1] L. Lenzini, E. Mingozzi, and G. Stea. Tradeoffs between low complexity, low latency, and fairness with deficit round-robin schedulers. *IEEE/ACM Transactions on Networking (TON)*, 12(4):693, 2004.

[2] J. Lockwood, N. McKeown, G. Watson, G. Gibb, P. Hartke, J. Naous, R. Raghuraman, and J. Luo. NetFPGA–an open platform for gigabit-rate network switching and routing. In *IEEE International Conference on Microelectronic Systems Education, 2007. MSE'07*, pages 160–161, 2007.

[3] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round-robin. *IEEE/ACM Transactions on Networking (TON)*, 4(3):385, 1996.

[4] G. Varghese and M. Shreedhar. Efficient fair queuing using deficit round robin. In *Proceedings of SIGCOMMâĂŹ95*. Citeseer, 1995.

[5] G. Watson, N. McKeown, and M. Casado. NetFPGA: A tool for network research and education. In *Workshop on Architecture Research using FPGA Platforms*. Citeseer, 2006.

[6] X. Zhang and L. Bhuyan. Deficit Round Robin Scheduling for Input-Queued Switches.