The Burst Crediting Concept

Sponsor: Sprint

Kert Mezger
David W. Petr

Technical Report TISL-10230-06

Telecommunications and Information Sciences Laboratory
Department of Electrical Engineering and Computer Sciences
University of Kansas

May 1995

Contents

1 Weighted Round Robin Server Background

It is the nature of asynchronous transfer mode (ATM) networks to provide flexibility in resource management. A network provider wants to have the ability to allocate bandwidth to customers according to their respective needs. This is the motivation for finding a resource allocation mechanism or algorithm that is dynamic enough to utilize the inherent flexibility of ATM. Many algorithms tend to take advantage of one or more characteristics of a network. These characteristics are usually related to the quality of service to the customer (e.g. the FIFO takes advantage of the timeliness of incoming traffic). The weighted round robin (WRR) resource allocation algorithm allows the network provider to allocate guaranteed bandwidth to customers based on the amount of time a particular queue has access to the server (i.e. transmission link) [1].

A WRR system consists of a number of input queues sharing access to a server. In applying WRR to a system, there are three basic conditions in the operation of the algorithm. First, if all the queues are empty, WRR waits for the next cell to enter any of the waiting queues. That queue will then be processed and given full access to the server. The second condition occurs when one queue is not empty and all the others are empty. Since no other queue needs access to the server, the filled queue is processed and will have access to the server until another queue is not empty. This leads to the third condition in which two or more queues are not empty. To resolve this contention, WRR resorts to using scheduling windows to determine queue access (see Figure 1). By starting at the beginning of the scheduling window and proceeding to the end, each scheduler slot is checked to see which queue will be given access to the server. If the queue associated with the current scheduler slot is empty, the next slot is checked immediately. When the last slot has been examined, the process starts over at the beginning [2].

Schedule #1 - Block

1	2	2	3	3	3	...	8	8	8	8	8	8	8	8

Schedule #2 - Distributed

8	7	6	4	1	3	5	2	4	8	5	7	6	1	...

Figure 1: Scheduling Windows

WRR uses these scheduling windows in order to guarantee proper bandwidth to each queue. When multiple queues are contending for the server, these scheduling windows show which queues will have access to the server, in what order, and for how long. Giving a particular queue more slots in the scheduling window allows more bandwidth to that queue. Likewise, fewer slots implies smaller bandwidth to a given queue. The guaranteed percentage of the link bandwidth afforded to a particular queue will be the number of slots allocated to that queue divided by the total number of slots in the scheduling window.

$$s = \text{number of slots allocated to queue N}$$

$$t = \text{total number of slots in the scheduling window}$$

$$\text{Guaranteed Percent BW} = \frac{s}{t}$$

Note, however, that a particular VC can obtain more than its guaranteed bandwidth if other VC's are not using all of their guaranteed bandwidth. Likewise, if a VC needs less than its guaranteed bandwidth, the difference can be used by other VCs. In this way, WRR provides service guarantees in a flexible manner. The ordering or arrangement of the slots in the window clearly will not affect the amount of bandwidth allocated to each queue [2]. However, delay should be dependent on the ordering of the slots in the window. Similarly, a longer scheduling window will not affect bandwidth as long as the percentage of the total slots allocated to each queue remains the same. The average VC (virtual circuit) delay using the short and long distributed scheduling window should be approximately the same due to similar wait times between servicing of a VC, while the average VC delay using the short and long block scheduling window should be different [1]. The long block scheduling window gives each VC a longer servicing time within a scheduling window, which creates a longer wait time between servicing of a VC. The distributed schedules used in this investigation were calculated based on a distribution algorithm developed by Q. Wang. This algorithm attempts to evenly distribute the slots for a given queue throughout the scheduling window [3]. The Matlab script file for this distribution is found in [1].

2 Burst Crediting Concept

The WRR algorithm provides guaranteed bandwidth to all VC's. In this respect, it performs well. If, on the other hand, a VC was to extend its quality of service criteria to include delay, some problems could occur. With a simple distributed window WRR server, a VC would have to have similarly distributed traffic to get a low amount of network delay. This is justifiable when the traffic is nonbursty (i.e. the ratio of the peak rate to the average rate is low), but when the traffic is burstier, many of the bursty cells experience a prolonged queue delay since the WRR server processes cells at the average rate (assuming heavy total server load). For data traffic, a simple WRR server may suffice, but a low average bandwidth image source sends a whole image worth of information at a time, expecting this image to be delivered with small delay. Simple WRR scheduling at the average image rate leads to a large network delay experienced by the image as a whole. Note that this is true whether the WRR schedule is block (potentially long wait until the block is serviced) or distributed (shorter wait for the first cell in the image burst, but long wait for the last cell in the burst). The challenge now is to find a way to serve the image traffic in a timely manner without violating the bandwidth agreements and without suffering a marked increase in delay to the other VC's.

The concept presented here is the idea of *burst crediting*. Burst crediting is essentially a loan given to a VC borrowed against its future bandwidth. Under "normal" conditions the system operates using a simple WRR mechanism. Once the queue of the bursty source reaches a certain threshold, however, the server goes into an interrupt mode which allows the bursty source exclusive use of the server. This is similar to many other high priority schemes [4]. The difference is that burst crediting puts a borrowing limit and a payback schedule into the algorithm. The borrowing limit is the maximum number of cells that a bursty VC can have served in interrupt mode without having to pay back credits. The payback schedule is a mechanism which forces the bursty VC to pay back the credits it borrowed in order to bring all VCs back to their guaranteed bandwidths. The payback schedule can repay the loan by skipping a schedule slot under "normal" WRR operations when the bursty queue is empty or filled. While empty, payback can occur every slot. While filled, payback occurs every *PB* scheduled slots.

The preliminary work presented below compares the classic WRR algorithm against the new WRR with burst crediting to see if the bursty source does encounter a lower average delay without drastically affecting the average delay of the nonbursty sources.

3 Simulations and Results

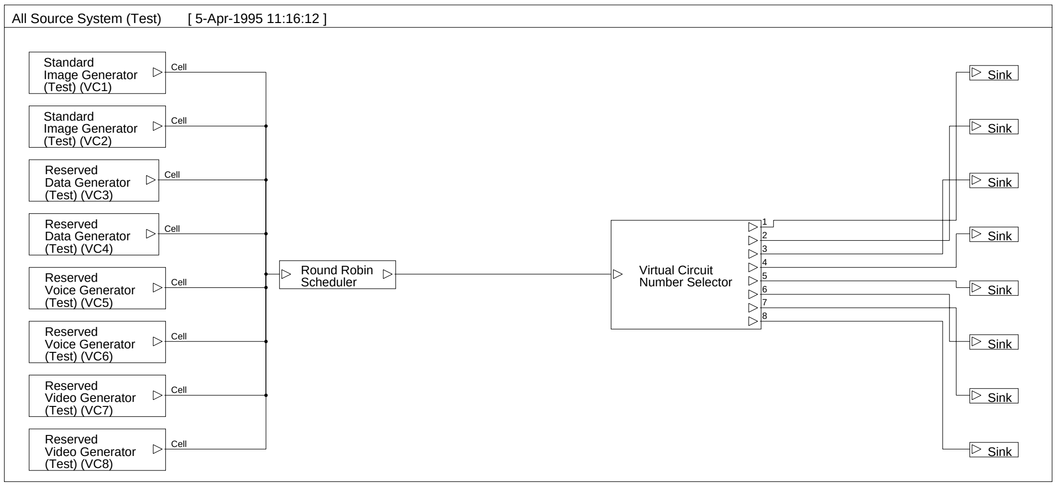

Figure 2: Network System BONeS Model

Using the BONeS network system in Figure 2, simulations were performed under 80, 90, and 100 percent loads for an output line capacity of 45 Mbps and 500-slot schedule length. The traffic sources and ATM cell segmenters are from previous research projects [3, 5]. The bandwidth allocated to each source as set by the schedule gave 30 percent to each data source, 7 percent to each voice and image source, and 6 percent to each video source. The allocation guaranteed sufficient average bandwidth to each source. The BONeS block diagram for the classic WRR mechanism is shown in Figure 3 and the block diagram for the WRR with burst crediting model is shown in Figure 4. Figure 5 shows that the image traffic does experience a significantly lower delay with a burst crediting server (mean delay improvement of nearly 25 milliseconds at 100% load). Figure 6 illustrates that the added delay experienced by the other traffic is minimal (mean delay penalty of about 5 milliseconds at 100% load). BONeS parameters for these simulations can be found in Table 1. These preliminary results encouraged further refinement and analysis of the burst crediting concept.

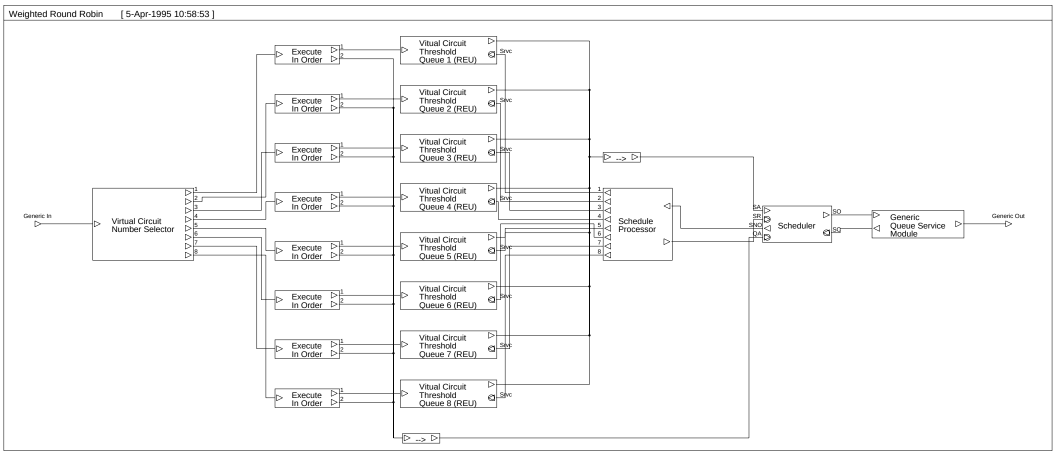

Figure 3: WRR BONeS Model

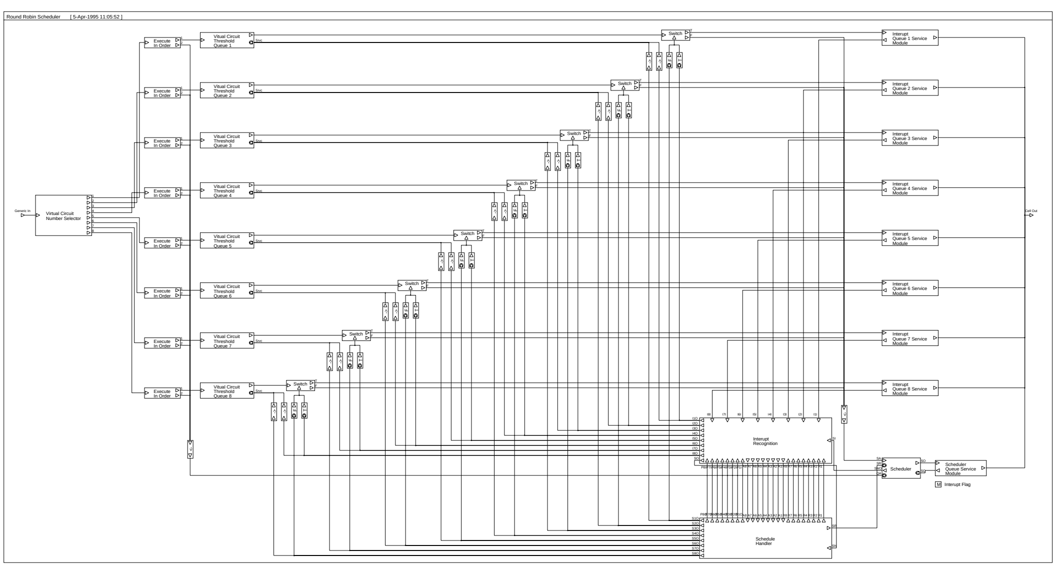

Figure 4: WRR with burst crediting BONeS Model

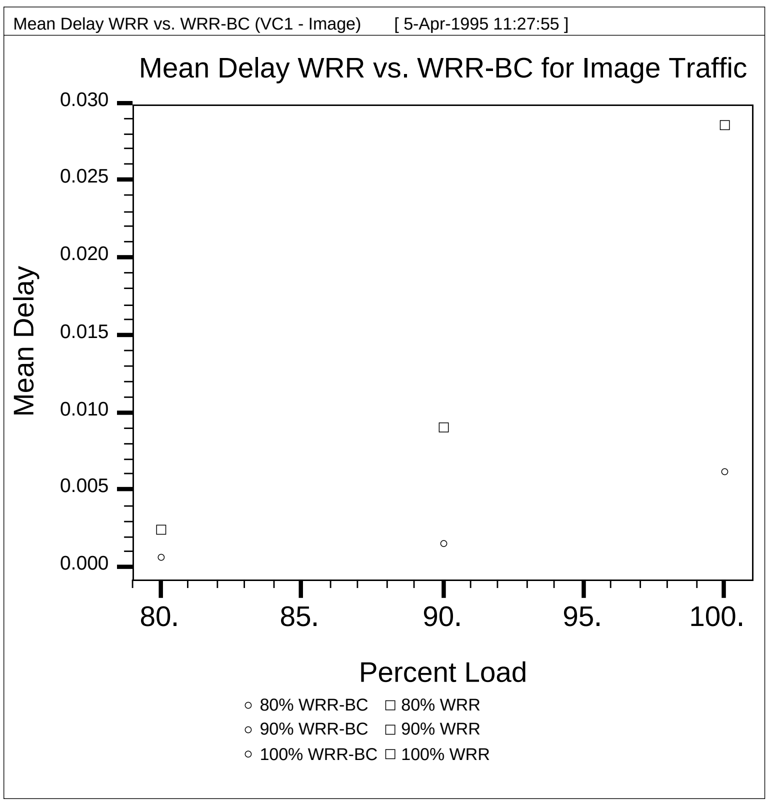

Figure 5: Mean Delay WRR vs. WRR-BC for Image Traffic

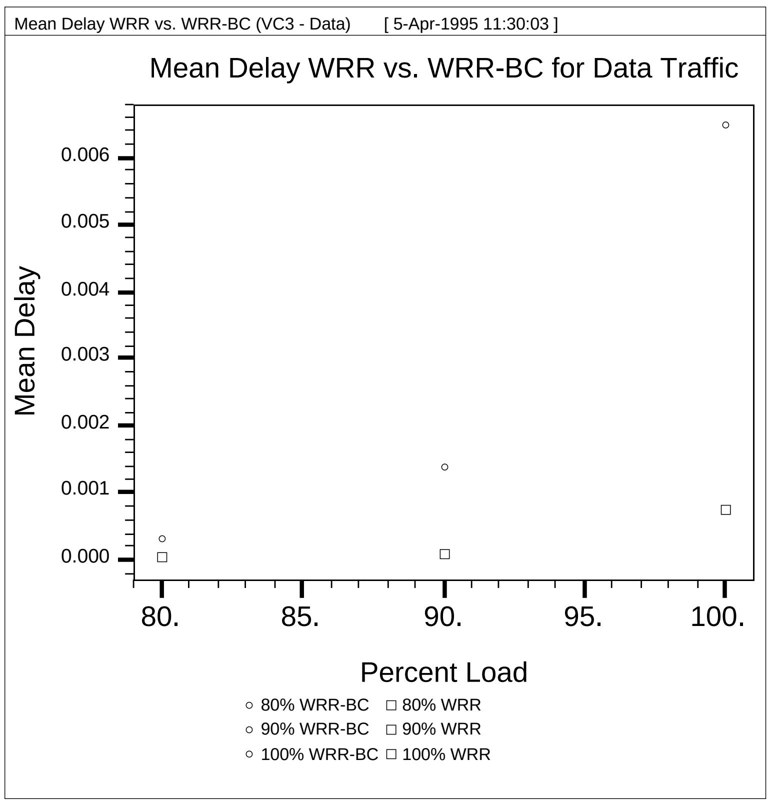

Figure 6: Mean Delay WRR vs. WRR-BC for Data Traffic

BONeS Parameter	*80 Percent Load*	*90 Percent Load*	*100 Percent Load*
Images per Hour	2542	2859	3177
Maximum Image Size (bits)	500000	500000	500000
Mean Image Length (bits)	400000	400000	400000
Average Bit Rate per Data VC	348160.0 * 27.0	391680.0 * 27.0	435200.0 * 27.0
Minlength (bits)	368.0	368.0	368.0
Maxlength (bits)	12000.0	12000.0	12000.0
Small Packet Exp. Mean (bits)	24.0	24.0	24.0
Large Packet Exp. Mean (bits)	3936.0	3936.0	3936.0
Probability of Small Packets	0.4	0.4	0.4
N (Number of Phones)	33	36	39
Voice Block Size (bytes)	42	42	42
Video Packet Length (bits)	728	728	728
Average Video Bit Rate (bps)	1700000.0	1912500.0	2125000.0
Maximum Queue Size	500	500	500
Peak Cell Rate (VC 1)	10E6	10E6	10E6
Peak Cell Rate (VC 2)	10E6	10E6	10E6
Peak Cell Rate (VC 3)	20E6	20E6	20E6
Peak Cell Rate (VC 4)	20E6	20E6	20E6
Peak Cell Rate (VC 5)	64000 * N	64000 * N	64000 * N
Peak Cell Rate (VC 6)	64000 * N	64000 * N	64000 * N
Peak Cell Rate (VC 7)	10E6	10E6	10E6
Peak Cell Rate (VC 8)	10E6	10E6	10E6

Table 1: BONeS Simulation Parameters

4 Conclusions

The preliminary results are sufficiently encouraging to justify additional algorithm refinement, analysis, and performance evaluation. Work on this new concept will confirm and be documented in subsequent technical reports.

References

[1] Brian Lang, Kert Mezger, David W. Petr. *Delay Performance of Weighted Round Robin Scheduling*. TISL Technical Report TISL-10230-03.

[2] Kert Mezger, David W. Petr, Timothy G. Kelley. *Weighted Fair Queuing vs. Weighted Round Robin: A Comparative Analysis*. IEEE Wichita Conference on Communications, Networks and Signal Processing 1994.

[3] David W. Petr, Q. Wang, Lyn Neir, Victor Frost. *Analysis and Simulations of Traffic Management Algorithms for Frame Relay/Fast Packet Networks*. TISL Technical Report TISL-9210-2.

[4] Duan-Shin Lee, Bhaskar Sengupta. *Queuing Analysis of a Threshold Based Priority Scheme For ATM Networks*. IEEE/ACM Transactions on Networking, Vol. 1, No. 6, December 1993.

[5] David W. Petr, Victor Frost, Ann Demirtjis, Cameron Braun. *Evaluation of Broadband Networking Technologies: Phase II Report*. TISL Technical Report

TISL-9750-4.