Current Trends in Technology and Science
ISSN : 2279–0535. Volume : 2, Issue : 1

Using Mini Round Robin to Provide Quality of Service For Multimedia Networks

A Kazmierczak
Computer information Systems, Northwest Arkansas Community College, One College Drive, Bentonville, AR 72712
akazmierczak@nwacc.edu

Abstract -There is a broad spectrum of packet data networks emerging onto the networking scene. The emergence of broad spectrum networks has also seen a growing number of applications involving multimedia communications. This growth in multimedia applications is a driving force towards providing a better Quality of Service (QoS) to users. A major component in providing QoS is the packet scheduler. In this paper we use a frame based scheduling technique known as Mini Round Robin (MRR) to provide a better QoS for multimedia applications. To the authors knowledge, there has been no previous work addressing the QoS scheduling of Diffserv traffic and packet transmission delay.

Keywords – Mini Round Robin, Packet Scheduler, Quality of Service

1. INTRODUCTION

Packet schedulers play a critical role in providing QoS guarantees in packet data networks. Some of the QoS guarantees include bounded delay, guaranteed bit rate and fair service allocation to all flows. This can only be achieved by solving the contention problem for shared resources and deciding on the sequence in which packets are transmitted from the node.

Modern packet scheduling theory is based on just a few concepts, though very important concepts. Generalized Processor Sharing (GPS) [1, 2] is the ideal scheduler that serves an infinitesimal amount of data from each flow according to reserved rate or relative bandwidth weight. GPS provides every flow its guaranteed bit rate and distributes excess bandwidth fairly among all flows.

Packet schedulers are classified into two main categories: time stamp based schedulers and frame based schedulers. In time stamp based schedulers, packets are time stamped upon arrival. The time stamp is subsequently used to determine the sequence in which packets are transmitted. This category includes Weighted fair Queuing (WFQ) [1, 3], Worst Case fair Queuing (WFFQ) [4], Virtual Clock (VC) [5] and Self Clocked Fair Queuing (SCFQ) [6]. Time stamped schedulers do provide tight latency bounds and provide good fairness. The major drawback of time stamped schedulers is their high work complexity, despite proposed improvements [7].

Frame based schedulers serve flows in rounds. In each round, a flow receives at least one transmission opportunity. Weighted Round Robin (WRR) [8], Deficit Round Robin (DRR) [9], and Elastic Round Robin (ERR) [10] are all frame based schedulers, Frame based schedulers are generally simple and have low work complexity.

In this paper, we restrict our attention to frame based scheduling based on Mini Round Robin (MRR). The rest of the paper is organized as follows. Section 2 provides an overview of common frame based schedulers. We focus on the MRR scheduler upon which our work is based. In section 3 we present our frame based scheduler that provides priority QoS to traffic based on the traffic classes proposed in the Diffserv [12]. We present our analytic performance of transmission delay times is section 4. In section 5, we present numerical results. Section 6 presents our conclusions.

2. PREVIOUS WORK

Round Robin (RR) is one of the simplest and most commonly used frame based schedulers, upon which so many frame based scheduling algorithms were subsequently based. In RR, backlogged flows are served in sequence, one packet at a time. All flows are treated equally. RR is considered fair if the same packet size is used for all flows and all flows have the same reserved rate. RR is simple and has a complexity of O91) per packet. However, in a more realistic environment where packet sizes are variable, as are session rates, RR does not perform well.

Weighted Round Robin (WRR) [8] was introduced to support flows with different reserved rates. WRR scheduling serves multiple packets from a flow based on the flow's normalized weight, where the weight of a flow is its relative share of total bandwidth. WRR shares RR's drawback in that using different packet sizes in different flows creates unfairness.

Deficit Round Robin (DRR) [9] overcame the unfairness caused by different packet sizes. In DRR, each flow iis assigned a quantum (Q_i). The quantum is proportional to the flow's weight and represents the ideal service the flow should receive in each round. Flows not using their quantum in a round get to transmit data in consecutive rounds. The quantum is added to the Deficit Counter (DC_i) of each flow at the beginning of a round. Packets are served as long as the flow has a positive DC. The quantum assigned should be greater, or at least equal, to the maximum packet size that could arrive in order for DRR to have a work complexity of O(1). If the quantum assigned is significantly higher than the maximum packet size, short term unfairness could occur. DRR requires knowledge of the packet size before scheduling. This piece of information may not be available for networks such as wormhole networks [10].

Current Trends in Technology and Science
ISSN : 2279–0535. Volume : 2, Issue : 1

Elastic Round Robin (ERR) [10, 11] introduced a variable quantum that depends on the performance in previous rounds. ERR allows a flow to exceed its allowance by one packet size. A Surplus Counter ($SC_i(r)$) tracks excess service in round r. After each round, the Maximum relative Surplus Counter (MaxSC(r)) is computed and used to calculate the new Allowance ($A_i(r)$) of flows to be served in the next round. This represents the least amount of data that can be sent in a round. ERR does not need to know the maximum packet size, yet provides better short term fairness and still maintains a work complexity of O(1) per packet.

Nested Deficit Round Robin (NDRR) [13] is another frame based scheduler. NDRR splits each DRR round into smaller inner rounds and runs a version of DRR in the inner rounds. The flow receives its quantum (Q_{min}) distributed over several inner rounds. Q_{min} is assigned to the flow with the lowest reserved rate. NDRR also needs to know the maximum packet size (M) that may arrive in order for NDRR to maintain a work complexity of O(1) per packet. If the packets that arrive are much less than M, NDRR displays unfairness. NDRR needs knowledge of the packet size before scheduling.

Pre-order Deficit Round Robin (PDRR) [13] and Prioritized Elastic Round Robin (PERR) [14, 15] are recent frame based schedulers. Bothe schedulers add a limited number of priority queues in which flows are sorted in ascending order of quantum size. PDRR classifies packets into priority queues according to the quantum availability. PERR sorts only the flow numbers into priority queues. PDRR is based on DRR while PERR is based on ERR. PDRR and PERR show improvement over DRR and ERR but have a work complexity of O(p) per packet, where p is the number of priority queues.

The most recently proposed frame based scheduler is the Mini Round Robin (MRR) [16] frame based scheduler. Since our scheduler is based on MRR, we describe MRR for reference.

MRR serves flows in rounds. A flow gets the chance to transmit a packet once every round. MRR divides each round into multiple mini rounds. In MRR, each flow is allowed to transmit one packet each mini round as long as it has packets to send. MRR does not require advance knowledge of maximum packet size.

MRR maintains two lists: ActiveFlowList and MiniRoundList. The ActiveFlowList keeps all flows with a non positive balance, while the MiniRoundList holds flows with a positive balance. At the start of a round, the contents of the ActiveFlowList are moved to the MiniRoundList. The flows in the MiniRoundList are served in order and one packet at a time. After each mini round, flows with non positive balance are excluded and a new mini round is started. When all flows become non positive, a new outer round is started with the flows balance updated according to equation 3 and a new series of mini rounds is started. $DC_i(r)$ is the deficit counter discussed previously.

$$DC_i(r) = Sent_i(r) - Balance_i(r)$$

$$MaxDC(r) = \max_{i \text{ served in } r}(DC_i(r) / W_i)$$

$$Balance_i(r) = W_i \times (1 + MaxD(r - 1)) - DC_i(r - 1)$$

After a packet is scheduled for transmission, the sessions reference number is appended to one of the lists as long as the flow has packets in its queue. The balance determines the list to which the flow is added. If its balance is positive, it goes to the MiniRoundFlowList, otherwise it goes to the ActiveFlowList.

3 Reducing The Transmission Delay Time For Diffserv Traffic

MRR lends itself to the idea of separate classes of traffic, especially traffic categorized by Diffserv [12]. Diffserv defines three classes of traffic: high priority, Expedited Forwarding (EF); medium priority, Assured Forwarding (AF); low priority, Best Effort (BE). Our focus is to reduce the delay seen by each class of traffic. We do this by providing transmission based on traffic class: EF transmitted first, AF transmitted second and BE transmitted last, regardless of the flow. We do this by keeping four flow lists: EFFlowList, AFFlowList, BEFlowList and MiniRoundFlowList. Using Diffserv traffic classes. The MRR sceduler woks as follows.

The QoS MRR scheduler consists of multiple nested mini rounds. The major outer round will process packets from all flows. The major outer round contains three major inner rounds. Each major inner round will process one traffic class. Thus there will be a major inner round for EF traffic, a major inner round for AF traffic and a major inner round for BE traffic. Each major inner rounds is divided into mini rounds, where each flow is scheduled.

Each round proceeds as follows. At the start of a major outer round, the EFFlowList is moved to the MiniRoundFlowList. The flows in the MiniRoundFlowList are served in order, one packet at a time just as in MRR. After each mini round, flows with a non positive balance are excluded and a new mini round is started. When all flows have a non positive balance, the next major inner round is started. The AFFlowList is moved to the MiniRoundFlowList. The flows in the MiniRoundFlowList are served in order, one packet at a time. After each mini round, flows with a non positive balance are excluded and a new mini round is started. When all flows have a non positive balance, the next major inner round is started. The BEFlowList is moved to the MiniRoundFlowList. The flows in the MiniRoundFlowList are served in order, one packet at a time. After each mini round, each flow with a non positive are excluded and a new inner round is started. When all flows have a non positive balance, the three major inner rounds are complete and a new major outer round is started. The flows balance is updated according to equation 3 given earlier.

After scheduling a packet for transmission, the sessions reference number is appended to the tail of one of the four flow lists as long as the flow has packets to transmit. The flows balance and traffic class determine the list to which the flow is added. If the balance is positive, the flow is added to the MiniRoundFlowList. If the flows balance is non positive, it is added to one of the lists according to its traffic class.

4. Analysis of Packet Transmission Delay

In this section, we provide a simplified analysis for transmission delay for EF, AF, and BE traffic packets. Let us establish notations used. Let N_E be the mean number of flows carrying EF traffic and let M_E be the mean number of EF packets per flow. The mean number of EF packets is then $E = N_E \times M_E$. Let T_E be the mean time to transmit an EF packet.

Let N_A be the mean number of flows carrying AF traffic and M_A be the mean number of AF packets per flow. The mean number of AF packets is $A = N_A \times M_A$. Let T_A be the mean time to transmit an AF packet. Let N_B be the mean number of flows carrying BE traffic and let M_B be the mean number of BE packets per flow. Let T_B be the mean time to transmit a BE packet.

There are two scenarios we address. In scenario 1, all packets have arrived for each flow and are available to be transmitted during the current round. For an arbitrary EF packet for an arbitrary flow E_j, the delay time until the packet is transmitted is

$$D_E = \sum_{i=1}^{j-1} (T_E \times E_i)$$

For an arbitrary AF packet, for an arbitrary flow A_j, the delay before this packet is transmitted is

$$D_A = \sum_{i=1}^{j-1} (T_A \times A_i) + D_E$$

For an arbitrary BE packet for an arbitrary flow B_j, the delay before the packet is transmitted is

$$D_B = \sum_{i=1}^{j-1} (T_B \times B_i) + D_A + D_E$$

The time for a complete major outer round is

$$T_{OR} = D_B + D_A + D_E$$

For simplicity, assume the time to transmit any packet is 2 time units and 50 packets for each traffic class. Then the total time for a complete outer round is 300 time units.

For scenario 2, we assume that during round r, new EF, AF and BE packets arrive but cannot be transmitted during round r and must wait until round r + 1. In this case, every packet must wait for at least one complete outer round. For an arbitrary EF packet for an arbitrary EF flow E_j the delay is

$$D_E = \sum_{i=1}^{j-1} (T_E \times E_i) + T_{OR}$$

For an arbitrary AF packet, for an arbitrary flow A_j, the delay before this packet is transmitted is

$$D_A = \sum_{i=1}^{j-1} (T_A \times A_i) + D_E + T_{OR}$$

For an arbitrary BE packet for an arbitrary flow B_j, the delay before the packet is transmitted is

$$D_B = \sum_{i=1}^{j-1} (T_B \times B_i) + D_A + D_E + T_{OR}$$

5. Numerical Results

In this section, we show numerical results for the transmission delay for EF, AF, and BE packets for scenario 1 and scenario 2. We consider packets at positions 10, 20, 30 , 40, and 50 in all three traffic classes. For ease of calculation we let the transmission time for all packets is 2 time units per packet. Our results are shown in table 1 for scenario 1 and table 2 for scenario 2.

6. Conclusions

In this paper, we presented a new scheduler that addresses QoS for Diffserv traffic classes. In multiple traffic scenarios, it is shown that higher priority traffic sees significantly lower transmission delay as compared to lower priority traffic.

Table 1 – Transmission Delays for Scenario 1

Traffic Class	Packet Number	Transmission Delay
EF	10	18
AF	10	118
BE	10	218
EF	20	38
AF	20	138
BE	20	238
EF	30	58
AF	30	158
BE	30	258
EF	40	78
AF	40	178
BE	40	278
EF	50	98
AF	50	198
BE	50	298

Table 2 – Transmission Delays for Scenario 2

Traffic Class	Packet Number	Transmission Delay
EF	10	318
AF	10	418
BE	10	518
EF	20	338
AF	20	438
BE	20	538
EF	30	358
AF	30	458
BE	30	558
EF	40	378
AF	40	478
BE	40	578
EF	50	398
AF	50	498
BE	50	598

Current Trends in Technology and Science
ISSN : 2279–0535. Volume : 2, Issue : 1

REFERENCES

[1] A. K. Parekh, R. G. Gallagher, "A generalized processor sharing approach to flow control in integrated services networks: the single node case", IEEE/ACM Transactions on Networking, Vol 1, pp 344 – 357, 1993

[2] S. Keshav, "An engineering approach to computer networking", Addison-wesley Publishing Co. 1997

[3] A. Demers, S. Keshav, S. Shenker, "Analysis and simulation of a fair queuing algorithm", Computer Communications Review SIGCOMM '89 Symposium Communications architectures and Protocols, Vol 19, no 4, pp 1 – 2, 1989

[4] J. C. R. Bennet, H. Zhang, "WF2Q: worst case weighted fair queuing", Proc IEEE Infocom, pp 120 – 128, Mar 1996

[5] L. Zhang, "Virtual Clock: a new traffic control algorithm for packet switching networks", ACM Transactions on Computer Systems, Vol 9 pp 101 – 124, May 1991

[6] S. Golestanic, "A self clocked fair queuing scheme for broadband applications", Proc IEEE Infocom, pp 636 – 646, 1994

[7] H. Tayyar, H. Alnuweiri, "The complexity of computing virtual time in weighted fair queuing schedulers", Proc IEEE ICC, Paris, 2004

[8] M. Katerenis, S. Sidipoulos, C. Courcoubetis, "Weighted round robin cell multiplexing in a general purpose ATM switching chip", IEEE Journal on Selected Areas of Communication, Vol 9, no 8, Oct 1991, pp 1265 – 1279

[9] M. Shreedhar, G. Varghese, "Efficient fair queuing using deficit round robin", IEEE/ACM Transactions on Networking, Vol 4, no 3, Jun 1996, pp 375 – 385

[10] S. S. Kanhere, H. Sethu, A. B. Parekh, "Fair and efficient packet scheduling using elastic round robin", Transactions on Parallel and Distributed Systems, Vol 13, no 3, pp 324 – 336, Mar 2002

[11] S. Kanhere, H. Sethu, "Low latency guaranteed rate scheduling using elastic round robin", Computer Communications, Vol 25, no 14, pp 1315 – 1322, Sep 2002

[12] S. Blake, D. Black, M. Carlson, E. Davis, Z. Wang, W. Weiss, "An architecture for differentiated services", IETF RFC 2475

[13] S. Tao, Y. Lin, "Pre order deficit round robin: a new scheduling algorithm for packet switched networks", Computer Networks, Vol 36, no 2-3, Feb 2001

[14] S. S. Kanhere, H. Sethu, "Prioritized elastic round robin: an efficient and low latency packet scheduler with improved fairness", Technical report DU-CS-03-03, Jul 2003

[15] S. Kanhere, "Design and analysis of faoir, efficient, and low latency schedulers for high speed packet switched networks", PhD Thesis, Drexel university, May 2003

[16] T. Al-Khasib, H. Alnuweiri, H. Fattah, V. C. M. Leung, "Mini round robin: an enhanced frame based scheduling algorithm for multimedia networks", Proc IEEE '05, Nov 2005, pp 363 - 368