Variably Weighted Round Robin Queueing for Core IP Routers

Yoshihiro Ito, Shuji Tasaka and Yutaka Ishibashi
Department of Electrical and Computer Engineering
Nagoya Institute of Technology
{yoshi, tasaka, ishibasi}@elcom.nitech.ac.jp

Abstract

This paper proposes a new lightweight QoS/CoS control method called VWRR (Variably Weighted Round Robin) for use on high-speed backbone networks. We evaluate the efficiency of VWRR by analysis and simulation. Queueing methods that can accurately guarantee the bandwidth like WFQ (Weighted Fair Queueing) usually put high processor loads; this makes them a poor choice for use on high-speed lines. In VWRR, the weight of the WRR (Weighted Round Robin) router varies adaptively depending on the average packet length; therefore, the router can adaptively control bandwidth allocation without drastically increasing the processing load. VWRR can trade the attained fairness for the processing load imposed. In an ideal case where no control delay exists, fairness of VWRR varies from that of WRR to the one by DRR (Deficit Round Robin). On the other hand, in an actual case where control delays exist, fairness of VWRR becomes better than WRR if the measurement interval is appropriately determined.

1. Introduction

Services and applications over TCP/IP networks have been diversifying along with the continuing expansion of the Internet. This means greater demands on the service providers for diversifying the *QoS (Quality of Service)*. In addition to providing a service that guarantees connectivity, new services such as *SLA (Service Level Agreement)* are also being demanded. The service level guarantee requires QoS control including traffic control at routers.

Traffic control schemes at routers are generally classified into two categories. One is queue length control such as *RED (Random Early Detection)* [1], and the other is packet scheduling like *WFQ (Weighted Fair Queueing)* [2]. Methods for packet scheduling require especially accurate measurement of IP packet length to control the amount of allocated bandwidth precisely.

Line speeds, on the other hand, are becoming high because of rapid advancement in processor performance and optical technologies such as DWDM. Gigabit Ethernet and 10 gigabit Ethernet over LAN/MAN as well as OC-48 and OC-192 lines over WAN, for example, can handle large capacities in gigabits per second. In order to transmit IP packets over these lines, the processor in a router must process millions of packets per second for each interface. Therefore, core routers with many interfaces in the backbone network have to process some tens or hundreds of millions of packets per second.

Consequently, the core routers cannot easily provide QoS or *CoS (Class of Service)* services using sophisticated scheduling methods such as WFQ, which imposes heavy processing loads on the routers because of bit-by-bit packet processing. This makes it preferable to use a simple scheduling method like Round Robin, which considers only the number of packets [3]. However, it is difficult for such a simple scheduling method to control the exact bandwidth over IP networks, which use variable length packets, because the amount of allocated bandwidth depends on packet length.

This paper proposes a lightweight method for controlling allocated bandwidth over IP networks. The method is referred to as *VWRR (Variably Weighted Round Robin)*, which utilizes three modules: a *WRR (Weighted Round Robin)* router, packet length observers and a weight calculator. In VWRR, the packet length observers measure the average packet length. Then, the weight calculator adaptively changes the weight of WRR according to the average packet length. This functional separation alleviates concentration of the router's load. Moreover, VWRR can trade the accuracy of guaranteed QoS for work load of the router according to the network environment.

0-7803-7371-5/02/$17.00 ©2002 IEEE

The rest of the paper is organized as follows. First, related works are described in Section 2. Then, VWRR is proposed in Section 3. Sections 4 and 5 make an analytical comparison and a comparison by simulation, respectively. Section 6 concludes the paper.

2. Related works

Many algorithms have already been proposed for packet scheduling. In this section, we describe three popular algorithms in relation to VWRR.

The most popular one among them is WFQ. It is currently implemented on many edge routers. Many algorithms similar to WFQ have also been proposed, e.g., see [4] and [5]. WFQ uses the bit slice concept and precisely guarantees the bandwidth and the maximum delay time. However, to guarantee bandwidth by WFQ, the scheduler within a router requires a large amount of packet processing because of packet sorting and others.

WRR is a simple scheduling algorithm that is supposed to be used for fixed cells or frames. The basic behavior of WRR is as follows. First, the classifier in the router puts each incoming packet into the queue of the corresponding flow. Next, the scheduler accesses each queue in a round-robin fashion. The maximum number of packets allowed to be transmitted from a queue in a round is specified by the weight for the queue. The weight is a predetermined constant integer according to the requested bandwidth. It should be noted that WRR does not work efficiently for networks with variable length packets, which have a fairness problem since WRR usually allocates the bandwidth on a packet-by-packet basis.

DRR (Deficit Round Robin) [6] is one of modifications of WRR to improve the fairness. DRR can guarantee fairness in terms of throughput. In DRR, the packet scheduler maintains a state variable called *DeficitCounter* to control the amount of dequeued traffic precisely. A quantity called Quantum is added to *DeficitCounter* in each round. If the length of the packet at the top of the queue is less than *DeficitCounter*, the packet scheduler can dequeue the packet and then subtracts the packet length from *DeficitCounter*. Otherwise, it accesses the next queue.

Although DRR requires only $O(1)$ processing work per packet in order to guarantee bandwidth, it must calculate the length of every packet. The calculation of the packet length takes some resources of the router. In particular, it is difficult to calculate the length of an IP packet encapsulated in MPLS [7] frame since the MPLS label stack does not include any length field [8].

3. VWRR

3.1 Basic concept of VWRR

In this section, we propose VWRR, which is based on WRR. Therefore, let us summarize the features of WRR before specifying VWRR. As pointed out in the previous section, WRR has a fairness problem. That is, even if two queues have the same weight, the bandwidth assigned to one queue may be halved if the packet length in the queue is half of that in the other queue. Therefore, the WRR scheduling is not suited for use over IP networks, which treat variable length packets. In order to improve the deficiency, the WRR router can define the weight in terms of bytes instead of packets; however, this paper adopts the WRR based on packet-units since the byte-unit one imposes more processing load on the router than the packet-unit one. Actually, some core routers that process a large number of packets per second use WRR scheduling based on packet-units as their traffic control method.

In order to fix the problem of WRR, VWRR changes the weight adaptively. It periodically calculates the average packet length for each queue rather than measuring the length of all packets, and the weight is computed with the average length. Calculation of the weight may be performed by any other entity than the scheduler so that the scheduling load does not increase. The administrator can also set the renewal time for a weight according to fluctuation in the average packet length, guaranteed bandwidth accuracy, and other factors.

3.2 Architecture

The VWRR method basically utilizes three modules: 1) *a WRR router*, 2) *packet length observers*, and 3) *a weight calculator*. Figure 1 illustrates the architecture of VWRR. The packet classifiers in the WRR router classify each incoming packet into a flow according to its service class and put it into the corresponding queue; the scheduler in the WRR router accesses each queue in proportion to the weight to allocate requested bandwidth. Also, each packet length observer periodically measures the average packet length for each flow on its input line, and it reports the average

length to the weight calculator. The weight calculator calculates the weight for each queue from the average packet length and the amount of the requested bandwidth; it then notifies the WRR router of each weight. Below we describe each module in detail.

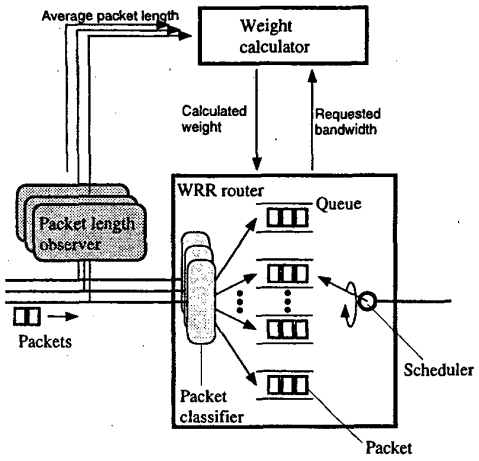

Figure 1. Architecture of VWRR

3.3 WRR router

The WRR router has packet classifiers, packet queues and schedulers. Although a scheduler usually exists for every output interface within the WRR router, Fig. 1 shows only one scheduler for simplicity. The packet classifiers classify incoming packets and puts them to appropriate queues. The scheduler accesses each queue on the basis of the WRR algorithm.

In VWRR, the method by which the weight calculator notifies the router of weight values is important. Many routers in the Internet support the standard TCP/IP management protocol, namely, SNMP [9], which uses the standard MIB [10] or vender MIB. Thus, the weight calculator can set the weight value for each queue in the WRR router with the standard method in SNMP; we do not need special techniques to implement the WRR router.

3.4 Packet length observer

The packet length observer classifies incoming packets and measures their length. It must determine to which flow each incoming packet belongs; it also computes the average length for each flow. If the router is capable of calculating the average packet length, then the packet length observer can be implemented inside the router.

Since a router generally has some input and output interfaces, it is necessary to place the packet length observer on each input line. Also, the weight calculator must collect information from all packet length observers.

In this paper, a packet length observer uses METER MIB [11], which is standardized by IETF RTFM WG [12]. With METER MIB, we can easily measure the average packet length of the flow corresponding to a queue according to user-defined rules. Moreover, since we can get METER MIB information with SNMP, the weight calculator can obtain the average packet length by means of SNMP messages.

Let us define the measurement interval as the period during which the packet length observer measures the packet length in order to obtain an estimate of its average. The measurement interval is a key design parameter that must be determined on the basis of fluctuation in the average packet length. If it is too short, the accuracy of the estimation becomes poor. On the other hand, for too large intervals the information obtained through the measurement becomes obsolete. That is, the best measurement interval depends on the degree of the packet length fluctuation, e.g., the variance of the packet length.

3.5 Weight calculator

The weight calculator calculates the weight coefficient of each flow from the average packet length measured by the packet length observers and the requested bandwidth. As the weight coefficient is calculated with the average packet length, VWRR can accurately allocate bandwidth regardless of difference in packet length between flows.

Now, let us suppose n flows. Then, the weight coefficient of queue i ($1 \le i \le n$) at time t, which is denoted by $w_i(t)$, is calculated by

$$w_i(t) = \frac{\max_{1 \le j \le n} p_j(t)}{p_i(t)} W_i \qquad (1)$$

where $p_i(t)$ is the average packet length of the flow corresponding to queue i at time t, $\max p_j(t)$ is the maximum average packet length at time t among the n queues, and W_i is the weight coefficient determined from the requested bandwidth for flow i by the WRR method; therefore, it is a constant integer. In order to remedy the unfairness of WRR due to variable length packets, Eq.

(1) magnifies W_i by the ratio of max $p_j(t)/p_i(t)$. In other words, if the average packet length for flow i is smaller than the maximum (i.e., max $p_j(t)/p_i(t)$ is larger than unity), then more than W_i packets in flow i can be transmitted in a round. If packet lengths of all queues are the same, the packet scheduler dequeues packets for W_i times in each round.

Since $w_i(t)$ means the number of packets which the packet scheduler is allowed to transmit from queue i in one access, it must be an integer. However, a very large $w_i(t)$ makes the round time extremely long. Therefore, $w_i(t)$ as a small integer is desirable. If $w_i(t)$ is calculated to be a large prime number as a result of calculating the average packet length, then dividing it by a constant common to all the flows and rounding off the result to a small integer will yield an adequate result. For example, let us suppose that there are three flows and that $w_1(t)$, $w_2(t)$ and $w_3(t)$ are calculated to be 101, 123 and 143, respectively. Dividing all the values by 20, we can get 5, 6 and 7 as more adequate $w_1(t)$, $w_2(t)$ and $w_3(t)$, respectively. However, since a large divisor to round off $w_i(t)$ causes large quantization error, we must not use very large divisors.

Actually, it takes some time for the packet length observer to report the average packet length to the weight calculator. Similarly, the weight calculator takes time to inform the weight. These delays can cause a problem. As described in the previous subsection, the packet length observer should be placed on each input line of the WRR router. It is preferable to place the packet length observers and the weight calculator near the input interface of the WRR router.

4. Analytical comparison of WRR, DRR and VWRR

In this section, we will make an analytical comparison of WRR, DRR and VWRR.

In order to evaluate fairness of scheduling algorithms, Shreedhar and Varghese [6] used a metric called *FairnessIndex*. The *FairnessIndex* for flow i ($1 \le i \le n$) is defined as

$$FairnessIndex_i = \frac{FQ_i \sum_{j=1}^{n} f_j}{f_i} \qquad (2)$$

where f_i is a quantity which expresses the ideal share to be obtained by flow i, and FQ_i is given by

$$FQ_i = \max\left(\lim_{t\to\infty} \frac{sent_{i,t}}{sent_t} \right) \qquad (3)$$

where $sent_{i,t}$ is the total number of bytes sent by flow i by time t, and $sent_t$ is the total number of bytes sent by all n flows by time t. The maximum in Eq. (3) is taken across all possible input packet size distributions for all flows.

As shown in [6], the *FairnessIndex* of WRR is Max/Min, where Max is the maximum packet size and Min is the minimum packet size. That of DRR is 1. The fairness of VWRR depends on the characteristics of traffic and some configurable parameters, for example, the measurement interval. In other words, we can flexibly set the fairness of VWRR according to some requirement. Therefore, it is very difficult to evaluate the fairness of VWRR with only one value. However, we can assess the best and worst values of VWRR fairness.

First, let us consider an ideal case in which the average packet length of each queue is precisely known in each round. We also suppose that the packet scheduler has rounded K times by time t. Here, we define $s_{i,k}$ as the number of bytes sent by flow i at the k-th round; then we have

$$sent_{i,t} = \sum_{k=1}^{K} s_{i,k} \qquad (4)$$

Also, let t_k denote the end time of the k-th round. Then, the average packet length of flow i at the k-th round, $p_i(t_k)$, is given by $s_{i,k}$ and the number of dequeued packets at the k-th round, $w_i(t_k)$, as

$$p_i(t_k) = \frac{s_{i,k}}{w_i(t_k)} \qquad (5)$$

From Eqs. (1) and (5), we may rewrite Eq. (4) as

$$sent_{i,t} = W_i \sum_{k=1}^{K} \max\left(p_j(t_k)\right) \qquad (6)$$

In order to consider the fairness, let us suppose that every flow requests the same amount of bandwidth. Then, we have

$$W_i = W \quad (i = 1,2,\ldots,n) \qquad (7)$$

where W is a constant integer. In this case, we may express FQ_i as

$$FQ_i = \max\left(\lim_{t\to\infty} \frac{W}{nW} \right) = \frac{1}{n} \qquad (8)$$

If f_1 through f_n have the same value, we have $FairnessIndex_i = 1$ ($1 \le i \le n$) from Eqs. (2) and (8).

Next, we consider the worst case, where we do not use the average packet length at all. Then, it is clear that the *FairnessIndex* takes the same value as that of WRR.

Table 1 shows the comparison of the *FairnessIndex* values among WRR, DRR and VWRR.

Table 1. Analytical comparison of fairness among WRR, DRR and VWRR

Queueing method	*FairnessIndex*
WRR	*Max/Min*
DRR	1
VWRR	[1, *Max/Min*]

5. Comparison by simulation

In the previous section, we showed an analytical comparison of WRR, DRR and VWRR. The *FairnessIndex* discussed in the previous section indicates how the fairness of one scheduling method is close to the one of the ideal bit-by-bit scheduling method, and this is a rigorous criterion. In this section, we consider only bandwidth as required service and evaluate actual bandwidth allocation of each method by simulation.

5.1 Simulation environment

We utilized BONeS Designer [13] as the simulation platform. Figure 2 depicts the simulation model. Three traffic generators issue packets. We call the three flows *flow 1*, *flow 2* and *flow 3*. For simplicity, we suppose that only one flow enters one input interface and that the number of output interface is one. This leaves out the classifier in the WRR router. The packet length information in each flow is collected by the corresponding packet length observer, which calculates the average packet length for each flow and reports it to the weight calculator at some constant interval. The weight calculator calculates the weight for each flow using the requested bandwidth and the corresponding average packet length.

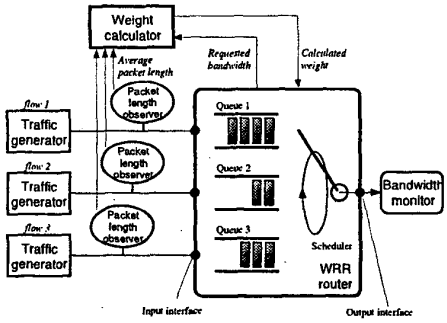

Figure 2. Simulation model

Each generated packet enters the WRR router and is assigned to the queue corresponding to the flow. The scheduler in the WRR router accesses each queue and dequeues packets whose number is given by the weight calculator. In our simulation we monitor the bandwidth allocated to each flow. We assume that the time it takes for the packet length observers to inform the weight calculator of the average packet length is zero. In addition, the time it takes for the weight calculator to inform the WRR router of the weight is assumed to be zero.

5.2 Traffic for performance evaluation

In the simulation, we used pseudo-traffic based on actual LAN data [14], which was adopted on the assumption that VWRR controls the QoS over trunk lines connecting LANs. Table 2 shows statistics of the LAN traffic.

Table 2. Statistics of each type of traffic

Average bit rate [kbps]	Average packet length [byte]	Standard deviation of packet length [byte]
1363	457	486

In the simulation, we generated pseudo-traffic according to the trace file of the actual traffic data. That is, each packet of pseudo-traffic has the same length and interval as those of the actual traffic. The three traffic generators generated the above traffic while changing the start location within the trace file with a uniformly distributed random variable. If the traffic generator reaches the end of the trace file during the simulation, it continues from the beginning of the trace file. The time duration of the trace file of the LAN traffic is 3142 [sec].

As described earlier, if the constant weight value W_i is large, the period in which the scheduler in the WRR router takes a round may become larger than the interval for renewing the weight. Conversely, if the value of W_i is small, the accuracy with which the bandwidth can be guaranteed declines. In this simulation, we reserved the same bandwidth for each queue for simplicity and selected W_i =6 (i=1,2,3). As a result of the simulation, we found that the total weight (i. e., $w_1(t)+w_2(t)+w_3(t)$) at any t had became 50 at most.

5.3 Metric for fairness

In the previous section, we showed the analytical comparison of each method using *FairnessIndex*. In Eq. (3), we must take the maximum across all possible input packet size distributions for all flows to calculate FQ_i. This means that the *FairnessIndex* is a rigorous criterion. Considering current services over the Internet, such strict fairness is not generally required. We should evaluate the fairness under actual traffic in a more realistic way. Then, we define a new metric based on Eq. (2) for actual traffic as follows. In order to make the calculation of the metric simple, we use the summations of the maximum consumed bandwidth and the total consumed bandwidth at each observation interval instead of taking the limiting value. That is, we define a metric q as

$$q = \sum_{m=1}^{N} \frac{\max_i B_m^{(i)}}{\sum_{i=1}^{n} B_m^{(i)}} \frac{\sum_{j=1}^{n} f_j}{f_i} \qquad (9)$$

where N is the total number of observation intervals, n is the number of flows and $B_m^{(i)}$ is the bandwidth consumed by flow i during the m-th observation interval. As the bandwidth for each queue is reserved more evenly, q approaches unity.

5.4 Simulation results

Table 3 shows parameters for our simulation. In Table 3, the observation interval is the interval to monitor the bandwidth allocated to each flow; thus, we have N=20 for the LAN traffic. Although it is preferable to use a higher speed link as the output line of a core router, we have selected the low speed link in order to make our simulation easy. However, results when we use a high-speed line will be estimated from this result.

Table 3. The simulation parameters

Simulation period [sec]	Output line speed [kbps]	Maximum buffer size [packet]	Observation interval [sec]
50	1500	100	2.5

Figures 3, 4 and 5 plot the amount of the carried pseudo-traffic as a function of time for the three flows when the measurement intervals are 50, 2 and 1 second, respectively. The packet length observers measure the lengths of all packets to calculate the average packet length. When the measurement interval is 50 seconds (i.e., the simulation periods), the weight value does not change from the initial value; therefore, the results in these cases are the same as those with the WRR scheduling. In addition, we present the simulation results of DRR in Fig. 6.

In Figs. 3 through 5, we observe that as the measurement interval becomes shorter, the fluctuation of the amount of the carried traffic decreases and the difference in the amount among the three flows also decreases. In Fig. 6, on the other hand, we notice that DRR achieves almost equal sharing of the bandwidth among the three flows.

Next, let us examine the effect of the measurement interval on the fairness of VWRR in terms of the metric q. Figure 7 plots the value of q as a function of the measurement interval. For comparison, the figure also shows the q values for WRR and DRR, which are constant independent of the measurement interval. A glance at Fig. 2 will reveal that the best measurement interval is 0.5 seconds for the LAN traffic. Thus, we see that the best measurement interval exists as already pointed out in Subsection 3.4.

From Fig. 7, we see that the q value of VWRR can be larger than that of WRR. As described in the previous section, if we can measure the exact average packet length, the *FairnessIndex* of VWRR becomes smaller than that of WRR. However, the average packet length of which the weight calculator is informed by the packet length observer is the value measured one measurement interval ago, not the current value. This difference between the current average packet length and the previous one degrades the accuracy of the bandwidth allocation. Therefore, it is necessary to use a more accurate estimation method of the average packet length to solve this problem. This is in future work.

Also, Fig. 7 shows that the value of q of VWRR approaches the one of WRR when the measurement interval is 50. Larger measurement interval decreases the frequency of the weight calculation. That is, as the frequency approaches zero, the performance of VWRR becomes close to that of WRR.

The packet length observer finds the packet length of the target flow and reports its average to the weight calculator. However, if it observes all packets, the load on the observer may become huge. Therefore, it does not observe all packets, but instead it does so at a regular sampling rate.

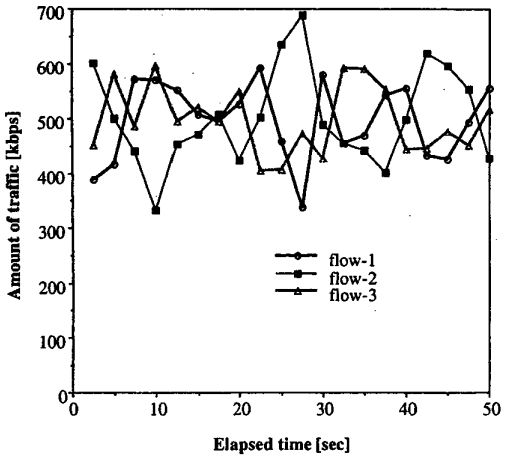

Figure 3. Progress of the amount of carried traffic (VWRR with a measurement interval of 50 seconds (WRR))

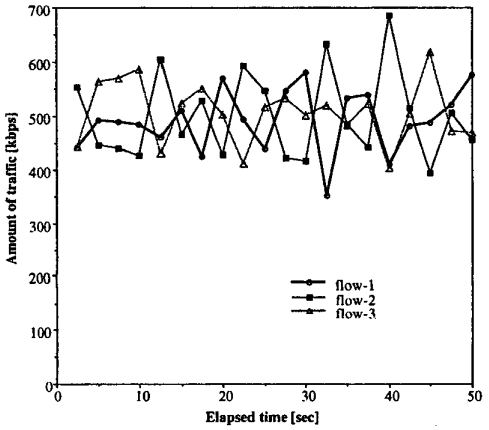

Figure 4. Progress of the amount of carried traffic (VWRR with a measurement interval of 2 seconds)

We now define the sampling rate $1/R$ as the rate at which a packet is observed for the calculation. That is, every R-th packet is observed. Then, let us consider the effect of the sampling rate on the accuracy of the guaranteed bandwidth. For that purpose, we measured the value of q under the same environment as before, by varying R from 1 to 100 packets. We used the LAN traffic for evaluation, and the measurement interval was set to 1 second. Figure 8 shows the result.

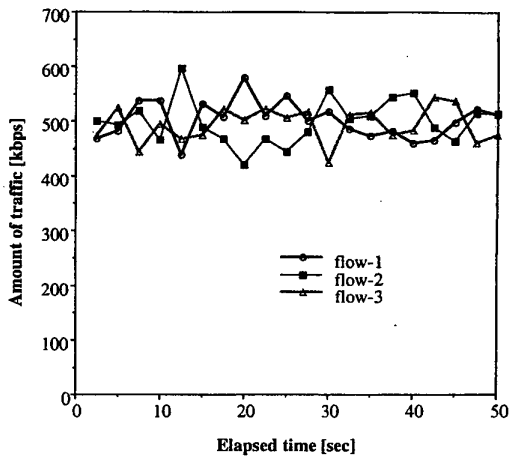

Figure 5. Progress of the amount of carried traffic (VWRR with a measurement interval of 1 second)

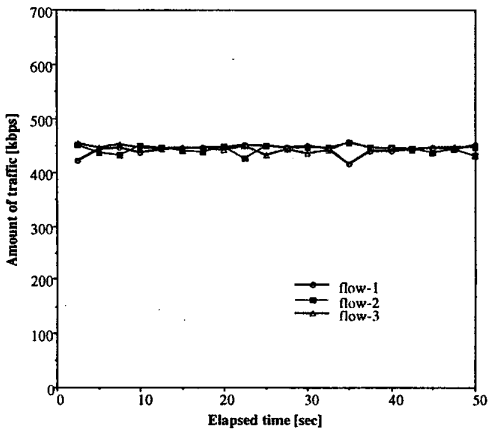

Figure 6. Progress of the amount of carried traffic (DRR for LAN traffic)

From Fig. 8, we see that the accuracy of the guaranteed bandwidth becomes almost the highest if R is smaller than 10 packets. Therefore, by setting the sampling rate to 1/10, the load on the packet length observer can be reduced to 1/10 of the load when all packets are measured without suffering a drop in bandwidth guarantee accuracy.

The above results indicate that the network administrator can set the sampling rate to the best value depending on the traffic characteristics of the target network and required accuracy of the guaranteed bandwidth. However, the method to select the sampling rate is future work.

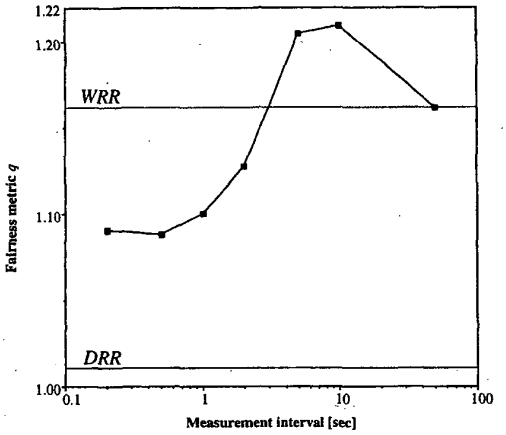

Figure 7. Fairness metric *q* for LAN traffic

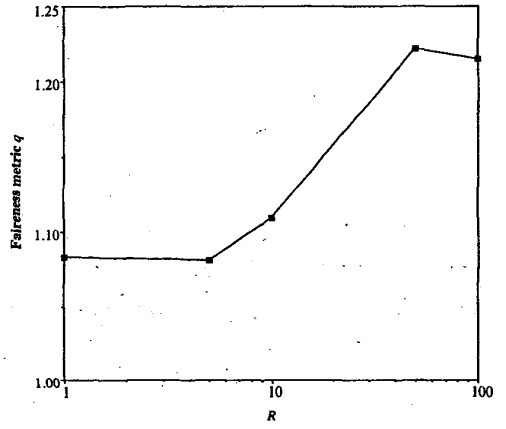

Figure 8. Fairness metric *q* versus *R*

6. Conclusions

This paper proposed a new lightweight scheduling method called VWRR to guarantee bandwidth over IP networks. In principle, VWRR can achieve all levels of the fairness ranging from that of WRR to the one by DRR. Moreover, by simulation, we showed that VWRR has better fairness than WRR when the measurement interval is small.

Since the major functions such as the packet length observer and the weight calculator are separated in VWRR, we can utilize existing WRR routers. Moreover, the network administrator can set VWRR parameter values so as to satisfy required accuracy of QoS control.

The study in this paper was done under a limited condition. Therefore, several subjects were left as future work. First, we plan to evaluate VWRR on ordinary routers that control QoS/CoS with the WRR algorithm and SNMP manager. Next, it is necessary to devise a more accurate estimation method of the average packet length utilizing the degree of the length fluctuation. Finally, we should discuss the best values of measurement interval and *R*. Although VWRR surely becomes better than WRR when the measurement interval and *R* is small, it is not clear how much we should decrease these values to get significant performance. We consider the following as one of such methods: The packet length observer adaptively changes the measurement interval and *R* while monitoring VWRR performance. The detailed algorithm for this method is future work.

References

[1] S. Floyd and V. Jacobson, "Random early detection gateway for congestion avoidance", *IEEE/ACM Trans. Networking*, vol.1, no.4, Aug. 1993, pp.397-413.

[2] A.Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm", *Proc. ACM SIGCOMM 89*, Austin, TX USA, Sept. 1990, pp.1-12.

[3] J. Nagle, "On packet switches with infinite storage", *RFC 970*, IETF, Dec. 1985.

[4] J. C. R. Bennett and H. Zhang, "WF2Q: Worst-case fair weighted fair queueing", *Proc. IEEE INFOCOM'96*, San Francisco, CA, USA, Mar. 1996, pp.120-128.

[5] S. J. Golestani, "A self-clocked fair queueing scheme for broadband applications", *Proc. IEEE INFOCOM'94*, Toronto, Canada, Apr. 1994, pp.636-646.

[6] M. Shreedhar and G. Varghese, "Efficient fair queueing using Deficit Round Robin", *Proc. ACM SIGCOMM 95*, Cambridge, MA USA, Aug. 1995, pp.231-242.

[7] E. Rosen, A. Viswanathan and R. Callon, "Multiprotocol label switching architecture", *RFC 3071*, IETF, Jan. 2001.

[8] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li and A. Conta, "MPLS label stack encoding", *RFC 3032*, IETF, Jan. 2001.

[9] J.Case, M. Fedor, M. Schoffstall and J. Davin, "Simple network management protocol", *RFC 1157*, IETF, May 1990.

[10] K. McCloghrie and M. Rose, "Management information base for network management of TCP/IP-based internets: MIB-II", *RFC 1213*, IETF, Mar. 1991.

[11] N. Brownlee, "Traffic flow measurement: Meter MIB", *RFC2064*, IETF, Jan. 1997.

[12] N. Brownlee, C. Mills and G. Ruth, "Traffic flow measurement: Architecture", *RFC 2063*, IETF, Jan. 1997.

[13] URL http://www.altagroup.com/alta/products/newdatasheets/designer.html

[14] URL http://ita.ee.lbl.gov/html/contrib/BC.html