

On maximum rate control of worst-case weighted fair queueing

Jeng Farn Lee
Yeali Sun

Department of Information Management,
National Taiwan University
E-mail: {r8725014,sunny}@im.ntu.edu.tw

Meng Chang Chen

Institute of Information Science,
Academia Sinica
E-mail: mcc@iis.sinica.edu.tw

Abstract-- Maximum rate control in a shared channel is important to service providers and carriers for various reasons. Previous approaches either use a concatenation of regulator and scheduler, which employs two set of queues and two management systems, or a policer in front of scheduler. The former requires extra buffer space and high overhead, and the latter causes inaccuracy. In this paper, we propose a new scheduling algorithm, called WF²Q-M (Worst-case Fair Weighted Fair Queueing with maximum rate control), to simultaneously support maximum rate control and provide minimum service rate guarantee. WF²Q-M employs a WF²Q like scheduler without policer or regulator that it is designed to provide accurate scheduling with low overhead. We prove that in WF²Q-M the packet's eligible time can be merged into its virtual starting time, and propose virtual clock adjustment to distribute the excess bandwidth of saturated sessions to other sessions without recalculating their virtual starting and finishing times. We also prove that WF²Q-M performance theoretically bounded by a fluid reference mode, and experiments show WF²Q-M performs just as claimed.

1. Introduction

Critical Internet applications have urgent performance requirements in terms of throughput, delay, delay jitter and loss rate, or a combination of above items. Current best-effort service model cannot meet the needs, as it handles all traffic equally and does not provide performance guarantees. A number of service disciplines endeavored to provide per-connection or per-queue performance guarantees [9], [1], [6]. Those disciplines provided minimum performance guarantees, but not maximum rate constraint, which is an important function for carriers, service providers, and applications. A few scenarios where maximum rate constraint application can be applied:

1. Control lease line's maximum services rate over concern that customers may use more bandwidth than their contacts. Customers must pay for higher service rate.
2. Restrict specific application's total outgoing traffic to enforce some management policies.

This work was partially supported by NSC under contract NSC 89-2219-E-001-006.

3. For some service classes like video or audio service classes, the sharing of bandwidth from the other non-backlogged classes cannot be used to admit extra requests, because the amount of shared bandwidth is not predictable. In addition, for some multimedia streaming applications, use of higher bandwidth (then designed) will overflow the receiving buffer that causes degraded service quality. As a result, the ban of excess bandwidth sharing for specific classes and applications is needed.

Consequently, many applications need service disciplines that it can simultaneously provide minimum performance guarantees and enforce maximum service rate constraint. In this paper, we propose a new service discipline called WF²Q-M (Worst-case Fair Weighted Fair Queueing with maximum rate control), which has the functions of WF²Q as well as maximum rate constraint capability. WF²Q-M is efficient in terms of buffer space, management complexity and computation cost..

2. Related Work

Several non-work conserving disciplines have been proposed. These include Jitter Earliest-Due-Date (Jitter-EDD) [11], Stop-and-Go [5], Hierarchical Round Robin (HRR) [8], and Rate-Controlled Static Priority (RCSP) [12]. They aimed to provide delay-jitter bounds, end-to-end delay bound or rate control based on either a time-framing strategy, or a sorted priority queue mechanism. In [13], Zhang and Ferrari showed that a general class of rate-controlled service disciplines could express all of them. As shown in Figure 1, a rate-controlled server has two components: a rate controller and a scheduler. The rate controller is responsible for shaping input traffic into desired traffic pattern. It assigns an eligible time for each packet, and moves the packets to scheduler when eligible. The scheduler multiplexes eligible packets from all connections. The rate-controlled service disciplines have two drawbacks:

1. More buffer space and complexity: The system needs two queues and two queue management systems for each session.
2. Additional operation cost: An important operation issue of rate-controller is when to move the packets to scheduler. The ideal method is to set a timer for each queue's head packet, but a timer introduces overhead

that is not acceptable for high-speed switches or routers. Most solutions in the literature are based on time-framing, event driven strategies or both. There is a tradeoff between system accuracy and time granularity in framing strategy. Smaller frame period means more accurate bandwidth allocation and higher operation cost, and larger frame period implies the opposite. Event-driven strategy is based on the occurrence of driving events, e.g. packet enqueue or dequeue, but, as the time of event occurrence is uncertain, high uncertainty is inherent in this approach. In any case, additional overhead is necessary to move packets in rate controller to scheduler and inaccuracy for rate-control occurs.

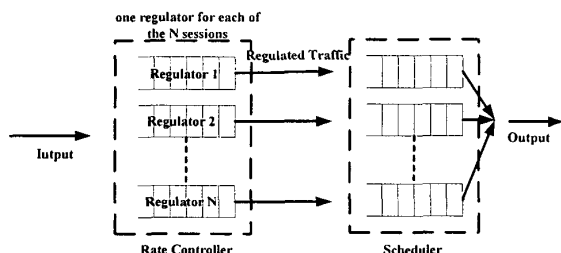


Fig. 1. Two-Stage Rate-Control Service Model

For the policer-based rate-control service model, as shown in Figure 2, token or leaky bucket [10] is used as policer, so that if an incoming packet obtains enough tokens, it moves directly to scheduler. Otherwise, it is dropped. Although this approach does not have the drawbacks described above, it has the following drawbacks: If the token bucket is used, the session's average rate can be maintained, but the token bucket allows burst traffic to be transmitted with a rate exceeding the designated maximum rate. If the rate controller uses leaky bucket, the maximum rate constraint can be strictly enforced, but bursty packets will be dropped.

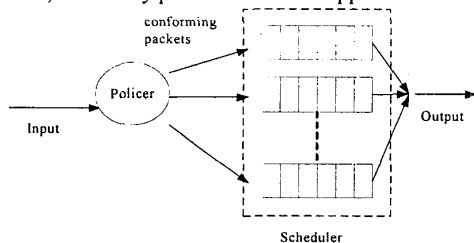


Fig. 2. Policer-Based Rate-Control Service Model

In this paper, we present a new service discipline without both rate-controlled and policer. Thus, the above drawbacks can be eliminated in the proposed service discipline.

3. WF²Q-M

In this session, we propose a new service discipline called WF²Q-M (Worst-case Fair Weighted Fair Queueing with Maximum Rate Control). WF²Q-M has only one set of queues and does not use policer, and is designed to provide accurate scheduling with low overhead. Like other weighted fair queueing services, users can define a set of sessions and specify a positive real number for each session. In addition, users can also assign maximum rates for some sessions, called maximum rate constrained sessions. WF²Q-M allocates bandwidth to the sessions according to their associated weights just like WF²Q, except the maximum rate constrained sessions receive at most their assigned maximum rates. For the non-maximum rate constrained sessions and sessions not receiving their maximum rates, they share the remained capacity according to WF²Q.

A. GPS-M Model

Now we introduce a fluid model GPS-M (Generalized Processor Sharing with Maximum Rate Control), which is an extension of GPS, as the reference model of WF²Q-M. In GPS, if the assigned weight of session i is ϕ_i , at time t , the shared bandwidth is $(\phi_i / \sum_{j \in B(t)} \phi_j) \times C$, where $B_p(t)$ is

the set of backlogged sessions at time t . If a maximum rate constrained session receives higher rate than its assigned maximum rate in GPS, we call the session is saturated, and the set of sessions at their peak rates is denoted as $B_p(t)$ at time t . (We also denote the set of sessions not in $B_p(t)$ as $\overline{B_p(t)}$.) In GPS-M, the sessions in $B_p(t)$ receive their peak rates and all others share the remained bandwidth as in GPS. When $\overline{B_p(t)}$ sessions receive the excess bandwidth from $B_p(t)$ sessions, some of them may get saturated, and the excess bandwidth must be distributed again. After the calculation, the resource allocation policy of GPS-M is represented as:

$$r_i(t) = \begin{cases} P_i & B_p(t) \\ \frac{\phi_i}{\sum_{j \in B(t)} \phi_j - \sum_{k \in B_p(t)} \phi_k} (C - \sum_{k \in B_p(t)} P_k) & \text{otherwise} \end{cases} \quad (1)$$

where $r_i(t)$ is the shared bandwidth of session i at time t . GPS-M service discipline is the same as GPS; it assumes that the server can serve all backlogged sessions simultaneously and that the traffic is infinitely divisible. The difference between GPS and GPS-M is that GPS serves session i with rate $(\phi_i / \sum_{j \in B(t)} \phi_j) \times C$, while

GPS-M serves with $r_i(t)$. GPS-M and GPS are exactly the same if $B_p(t)$ is null. As the definition above, GPS-M may be work-conserving or non-work conserving; when all the backlogged sessions are in $B_p(t)$ and the sum of the sessions' maximum rate is less than the link capacity, GPS-M becomes non-work conserving.

B. Virtual Clock Adjustment

Here we proposed a mechanism *Virtual Clock Adjustment* to distribute the excess bandwidth from sessions in $B_p(t)$ to the sessions in $\overline{B}_p(t)$ according their assigned weights. An intuitive method is to recompute virtual starting and finishing times of all packets in the system when backlog changes. But this method is infeasible due to high overhead. Therefore, we propose a new method adjusting the ticking rate of virtual clock in order to enforce the correct bandwidth distribution to sessions.

The following example shows how virtual clock adjustment works. There are four sessions sharing the same link. For simplicity, assume that all packets have the size of 1, and link speed is 1. Also, let reserved bandwidth weights of four sessions be 50%, 25%, 12.5% and 12.5%, respectively. The second session has maximum rate constraint of 0.4. The first session becomes inactive at time 0 while each of the remaining sessions sends 1 packet at the beginning of very second. For WF²Q, the virtual starting and finishing times of packets are shown in Figure 3, and the service order is shown in Figure 4. The transmission rates of the sessions 2, 3, 4 are 0.5, 0.25 and 0.25, respectively.

Since the maximum rate of session 2 is 0.4, if there is maximum rate enforcement, the excess 0.1 bandwidth will be distributed to sessions 3 and 4. The resulting bandwidth distributed to sessions 2, 3 and 4 are 0.4, 0.3 and 0.3. As shown in this example, sessions in $\overline{B}_p(t)$ may benefit from the excess bandwidth of $B_p(t)$. In a weighted fair queueing environment, virtual starting time and finish times of packets determine the transmission sequence in a shared channel, as well as the bandwidth distribution. For instance, the first packet of session 3 finishes service at virtual clock 8, which is 4 of real clock in WF²Q. While in WF²Q-M, session 3 is allocated bandwidth 0.3, instead of 0.2, and, thus, its finish time in real clock should be moved forward. Instead of modifying the virtual starting and finish times, WF²Q-M modifies the mapping between real clock and virtual clock to reflect the bandwidth increase in $\overline{B}_p(t)$. The real clock to virtual clock

$$\text{mapping ratio is } \sum_{j \in B(\tau)} \phi_j * \frac{\frac{\sum_{j \in B(\tau)} \phi_j - \sum_{k \in B_p(\tau)} \phi_k}{\sum_{j \in B(\tau)} \phi_j} * C}{C - \sum_{k \in B_p(\tau)} P_k}$$

Where $\sum_{j \in B(\tau)} \phi_j$ is the original ratio, the numerator is the original total bandwidth shared by $\overline{B}_p(t)$ sessions, and the denominator is the total bandwidth shared by $\overline{B}_p(t)$ sessions after receiving excess bandwidth from $B_p(t)$ sessions. For instance, $F_3^1 = 8$, which maps to 4 of real clock in WF²Q, while in WF²Q-M, the real clock is 10/3 (i.e. $8 * (0.5 * (0.5/0.6))$). To summarize,

$$\text{ratio}(t) = \begin{cases} \frac{\left(\frac{\sum_{j \in B(\tau)} \phi_j - \sum_{k \in B_p(\tau)} \phi_k}{\sum_{j \in B(\tau)} \phi_j} \right) * C}{C - \sum_{k \in B_p(\tau)} P_k} & \text{if } \sum_{j \in B(\tau)} \phi_j - \sum_{k \in B_p(\tau)} \phi_k \neq 0 \\ \sum_{j \in B(\tau)} \phi_j & \text{Otherwise} \end{cases} \quad (2)$$

The system virtual clock $V(t)$ of WF²Q-M evolves as following:

$$V(0) = 0$$

$$V(t_{j-1} + \tau) = V(t_{j-1}) + \frac{\tau}{\text{ratio}(t)} \quad (3)$$

25% session 2				
Virtual start time	0	4	8	12
Virtual finish time	4	8	12	16
maximum rate 0.4				
12.5% session 3				
Virtual start time	0	8	16	24
Virtual finish time	8	16	24	32
12.5% session 4				
Virtual start time	0	8	16	24
Virtual finish time	8	16	24	32

Fig. 3. Virtual starting and virtual finishing times of packets in WF²Q

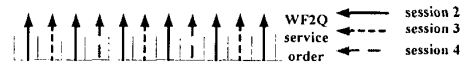
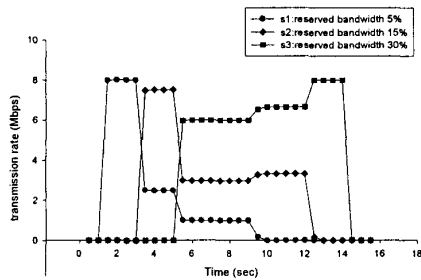


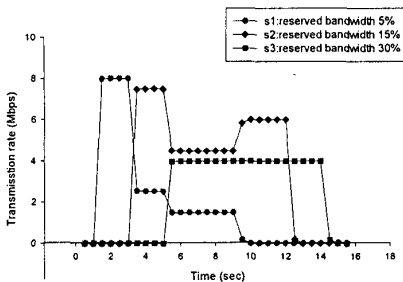
Fig. 4. WF²Q Service Orders

C. WF²Q-M Model

To achieve maximum rate control for $B_p(t)$ sessions,



8-a. WF^2Q



8-b. WF^2Q-M

Fig. 8. Four connections with reserved bandwidth 1:3:6:10. Session 3 has maximum rate control of 4Mbps. Session 4 is inactive.

5. Discussions and Conclusion

In this paper, we propose a new service discipline WF^2Q-M that it can guarantee minimum service rates as WF^2Q and provide maximum service rate constraint simultaneously. This paper shows that packet's eligible time can be merged into its virtual starting time to reduce complexity. The virtual clock adjustment allows the sharing of excess bandwidth without recomputing virtual starting and finishing times of packets. In addition, WF^2Q-M takes advantage of the nice properties of WF^2Q for weighted fair queuing. We also show that the performance of WF^2Q-M is theoretically bounded by a fluid reference model, similar to WF^2Q bounded by GPS model.

References

[1] J. C. R. Bennett and H. Zhang, "WF2Q: worst-case fair weighted fair queuing," in Proc. IEEE

INFOCOM'96, San Francisco, CA, Mar. 1996

[2] J. C. R. Bennett and H. Zhang, "Hierarchical packet fair queueing algorithms," IEEE/ACM Trans. Networking, vol. 5, pp. 675-689, Oct. 1997.

[3] D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide-area networks." IEEE Journal on Selected Areas in Communications, 8(3):368-379, April 1990

[4] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," IEEE/ACM Trans. Networking, vol. 3 pp. 365-386, Aug. 1995.

[5] S. Golestani, "Congestion-free transmission of real time traffic in packet networks." In Proceedings of IEEE INFOCOM'90, pages 527-542, San Francisco, California, June 1990.

[6] S. Golestani, "A self-clocked fair queueing scheme for broadband applications." In Proceedings of IEEE INFOCOM'94, page 636-646, Toronto, CA, June 1994

[7] UCB/LBN/VINT. Network simulator ns(version 2.1b6) <http://www.isi.edu/nsnam/ns/>, 1999

[8] C. Kalmanek, H. Kanakia, and S. Keshav, "Rate controlled servers for very high-speed networks." In IEEE Global Telecommunications Conference, pages 300.3.1 – 300.3.9, San Diego, December 1990.

[9] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case", IEEE/ACM Transactions on Networking, Vol. 1, No. 3, pp.344-357, June 1993.

[10] A. S. Tanenbaum, "Computer Networks", 3rd edition, Prentice-Hall, Inc. p380-381, 1996

[11] D. Verma, H. Zhang, and D. Ferrari, "Delay Jitter Control for Real-Time Communication in a Packet Switching Network." In Proceedings of Tricomm'91, pages 35-46, Chapel Hill, North Carolina, April 1991

[12] H. Zhang and D. Ferrari, "Rate-controlled static priority queueing." In Proceedings of IEEE INFOCOM'93, pages 227-236, San Francisco, California, April 1993.

[13] H. Zhang and D. Ferrari, "Rate-controlled service disciplines." Journal of High Speed Networks, 3(4):389-412, 1994

[14] H. Zhang, "Service Disciplines for Guaranteed Performance Service in Packet-Switching Network", Proc. IEEE, Vol. 83, October 1995, pp. 1374-139