PRIORITY-BASED NETWORK QUALITY OF SERVICE

ANIMESH DALAKOTI, NINA PICONE, BEHROOZ A. SHIRAZ[†]
School of Electrical Engineering and Computer Science
Washington State University, WA, USA 99163

WEN-ZHAN SONG†
School of Engineering and Computer Science
Washington State University-Vancouver, WA, USA 98686

ALI HURSON†
Department of Computer Science and Engineering
Pennsylvania State University, PA, USA 16802

In this paper we propose a Variable-Weighted Fair Queuing scheduling algorithm (V-WFQ) as an alternative to Weighted Fair Queuing (WFQ) that provides dynamic adjustments to message priorities as network congestion changes. Our results show that we are able to provide better Quality of Service to messages with higher priorities when compared with WFQ.

1. Introduction

Providing Quality of Service (QoS) [1][2] in the Internet has been a significant research area in recent years. QoS provisions can be achieved through the functioning of several network entities which control the buffer and bandwidth allotment to competing traffic flows, such as scheduling algorithms. Scheduling algorithms control one way end-to-end delay, throughput, packet loss, and bandwidth management by deciding which packet to forward next from several queues containing traffic from different incoming flows. The Diffserv [3] QoS architecture, defined by the Internet Engineering Task Force (IETF) to provide service differentiation among Internet traffic, uses Fair Queuing [2] and WFQ [1] scheduling algorithms. WFQ is a static scheduling algorithm, in which every flow/queue is associated with a fixed weight and packets are scheduled based on the Type of Service (ToS) level priorities. We propose a congestion-reactive V-WFQ algorithm which is a dynamic QoS provisioning scheduling algorithm,

---

[†] This work is partially supported by NASA AIST Grant #106269, NSF-ITR grant IIS-0326505, and NSF-ITR grant IIS-0324835.

allowing it to be deployed within the current Diffserv architecture. Unlike WFQ, which has a one to one static mapping, V-WFQ has a dynamic mapping which changes with the current congestion.

The rest of the paper is organized as follows: Section 2 defines the architecture of V-WFQ, Section 3 elaborates on the details of the V-WFQ algorithm, and Section 4 presents the results of our simulations and comparisons between V-WFQ and WFQ. Finally, the conclusions are discussed.

2. The V-WFQ Algorithm

Our proposed congestion-reactive V-WFQ algorithm has been designed to assist the Diffserv [3] architecture in providing Quality of Service performance improvements for high priority messages. We assume the packets have four priority levels based on their type of service consistent with the Diffserv architecture: Emergency services, Paid Premium services, Administrative services, and General services. These services are processed in order of their priority within the Diffserv architecture. Thus, Emergency services have the highest priority while General services have the lowest priority.

For a network router let *nQueue* be the number of queues at the router’s scheduler and *nToS* be the number of ToS levels. In WFQ, *nQueue* is always equal to *nToS*. However, in V-WFQ, *nQueue* is always greater than *nToS*. The additional queues in V-WFQ are used to push low priority messages into lower priority queues when network congestion levels are high. The idea is that high priority messages remaining in higher priority queues will have a better chance of being serviced compared to the low priority messages. The algorithm is implemented using an 8-bit field in the packet header called P-bits (for priority bits). For a given message, its P-bits value will dynamically change as a packet is forwarded from one router to the next according to the network congestion.

2.1. *Initialization Phase*

When a packet is sent out from a source, it’s P-bits (8-bit field) in the message header is set to all 1's. An all 1’s P-bits value indicates a situation in which there is no or low levels of congestion on the communication link. When the P-bits are set to all 1's, V-WFQ behaves the same as WFQ in scheduling a router's incoming queues. When a packet arrives at a router, its P-bits will be changed according to the congestion level on the router's output link. Thus, through the manipulation of the P-bits we will keep higher priority packets in higher priority queues while pushing lower priority packets to even lower priority queues as explained in the following section. When the message is put on the output link of a router, its P-bits will be reset to all 1's. Thus, P-bits will dynamically change as a packet moves from source to destination, effectively changing the priority of the packet according to the congestion levels at each router. The priority of a packet is set by the service application using ToS-bits (3 bits for Type of Service). The Type of Service and corresponding ToS-bits are listed in descending order of importance: Emergency Services (111), Administrative Services (110), Paid Premium Services (101), and General Services (100).

2.2. *The Architecture*

The system architecture consists of four modules: the Shifter, the Scheduler, the Priority Look-up Table (PLT), and the Classifier. The V-WFQ algorithm introduces the Shifter and PLT as new modules within its QoS architecture. The PLT is essential to V-WFQ as it is responsible for determining how the shifter shifts the P-bits whenever a new packet arrives at a router. In WFQ with 4 ToS levels, there will be 4 queues and the ToS value will determine which queue a packet is assigned to. In V-WFQ, however, more input queues are utilized at each router (as determined by the network administrator). In our scenario, in the PLT it is assumed we have 6 queues for the 4 ToS, labeled as P, P/2, P/4….P/32, with queue P having the highest priority and P/32 having the lowest priority. When a packet arrives at a router, its P-bits value will be right-shifted to lower its priority according to the congestion situation on the output link of the router. For example, when a packet of ToS = 111 arrives and the congestion level is at 85%, then its P-bits is right-shifted twice and the packet is assigned to the P/4 queue. On the other hand, if the congestion level is at 65%, the packet will be right-shifted once and assigned to the P/2 queue.

The PLT must be defined by the network administrator. The right-shifted value of the P-bits (from the PLT) determines the queue to which each packet will be scheduled. The PLT defines the mapping from which the P-bits should be right-shifted based on the current congestion level at the router and the packets ToS. As congestion increases, higher level ToS flows are designated to queues with higher priorities while lower level ToS flows are shifted to queues with lower relative priorities ensuring that as congestion increases, higher level ToS flows are given an increased share of the network resources.

The classification table is used by the classifier to allocate packets based on their new (shifted) P-bit values to the input queues at each router. Once the packets are classified into the input queues, they will be processed based on the WFQ algorithm in which packets are sent from the queues to the router on a round-robin basis. However, queues are weighted such that more packets are processed from the higher priority queues at each turn. From our example, the first 25 packets are processed from the P queue, then 15 packets from the P/2 queue, then 9 packets from the P/4 queue, and so on. As congestion increases higher relative weights are assigned to the higher level ToS flows resulting in better performance for these flows.

3. Putting it all together: V-WFQ Algorithm

V-WFQ has been designed as an independent scheduling algorithm which can be used to enhance the Diffserv architecture. Its flow of operations is shown in Figure 1. When a packet starts at the source, all of its P-bits are set to 1. In addition, the ToS bits are determined by the ToS level corresponding to the flow. Once the packet reaches a router, the shifter module uses the PLT table to determine the appropriate value of the P-bits based on the current congestion level within the network and right-shifts the P-bits accordingly. V-WFQ requires the knowledge of the current congestion (specified as a percentage) at the router, computed as a function of the backlogged, or non-empty, queues of the scheduler. Next, the packet is forwarded to the classifier, which assigns it to a specific queue depending on the new P-bits value. A WFQ scheduler schedules the packets from each queue according to its rate, which is proportional to the weight assigned to that queue. After a router processes a packet, it is sent to the desired port and its P-bits are again set to 1.

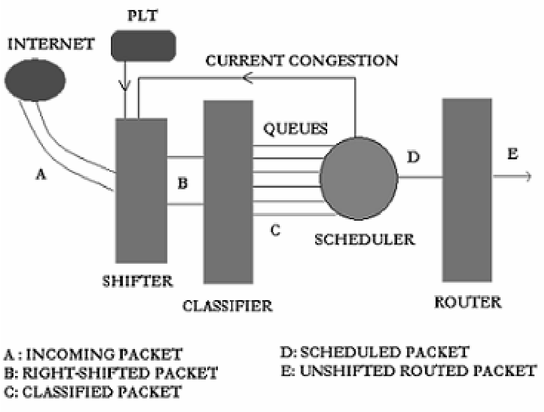

Figure 1: Control flow and system architecture.

Using V-WFQ, as network congestion increases, the gap in the percentage of resources being assigned amongst the different ToS level flows is widened, favoring the higher level ToS flows. Thus, as congestion increases, more resources are allocated to the higher level ToS flows. In contrast, with WFQ, the percentage share of network resources amongst traffic flows remains constant throughout the lifetime of the network regardless of network congestion.

4. Performance Evaluation of V-WFQ

This section presents performance comparisons between V-WFQ and WFQ based on NS-2 simulations. Our network consists of 4 servers represented as 4 source machines, S1-S4, connected to 8 user machines, U1-U8, through 3 routers, R0-R2 arranged in a ring topology.

Our experiment scenario represents a random, bursty communication pattern in which network traffic is unevenly generated by the servers in a non-periodic manner. This experiment captures the scenario of a particular ToS level traffic consumes the system resources by dominating the system (e.g., video streaming of medical data).

We compared V-WFQ and WFQ, using the following metrics: average packet end-to-end delay, throughput, percentage of packet loss, and an aggregate performance metric called Weighted Average System Delay (WASD). The average packet delay represents the average of all the packet end-to-end delays given a system architecture, a network load, a network traffic pattern, and a time duration. While the average packet delay is a useful metric for comparing two different systems, it does not reflect the priority of the packets. For example, the average packet delay may be the same for two different systems, but one system may provide a better quality of service for high priority packets than the other. WASD is designed to represent the system average packet delay while taking into account the relative importance among the packets according to their priority level. Let APD_i represent the average packet end-to-end delay for all packets of ToS level i (or priority level i) and let RW_i be the relative weight of ToS level i compared to all other ToS levels. For example, if the relative weight of ToS level 111 is 0.56 while the relative weight of ToS level 101 is 0.11, this indicates that ToS level 111 is 5 times more important in this system compared to ToS level 101. For n ToS levels, we have:

$$\text{Average packet delay} = \left(\sum_{i=1}^{n} APD_i\right)/n \qquad (1)$$

$$\text{WASD} = \sum_{i=1}^{n} RW_i * APD_i \qquad (2)$$

4.1. *Experiment Results*

Figures 3 (a) through (d) compare the performance of WFQ vs. V-WFQ over 10 runs of the experiment. Figure 3 (a) depicts the average packet end-to-end delay for each type of service. Compared to WFQ, V-WFQ provides a better end-to-end packet delay for high priority packets (ToS 111 and 110) while performing poorer for low priority packets, especially for packets of ToS 100. Figure 3 (b) shows the packet drop rate between WFQ and V-WFQ. As expected, V-WFQ performs better for high priority packets compared to low priority packets, but it is interesting to note that the packet drop rate for the lowest priority packets is almost the same for V-WFQ and WFQ. Similarly, in Figure 3 (c) V-WFQ and WFQ perform comparably in terms of throughput. V-WFQ outperforms WFQ for high priority packets in terms of end-to-end packet delay, packet drop rate, and throughput. For low priority packets, V-WFQ performs worse than WFQ in terms of end-to-end delay, but comparably in terms of packet drop rate and throughput due to the use of additional queues to keep the low priority packets in the network when the congestion level goes up. Finally, Figure 3 (d) shows the percentage gain by V-WFQ over WFQ based on the WASD metric when using different relative weights for the ToS levels. The WASD for the 4 different scenarios are listed according to the observed relative weights for each ToS level in descending order: W1= 0.53, 0.27, 0.13, 0.07; W2= 0.82, 0.10, 0.05, 0.03; W3= 0.75, 0.15, 0.08, 0.02; and W4= 0.76, 0.19, 0.04, 0.01.

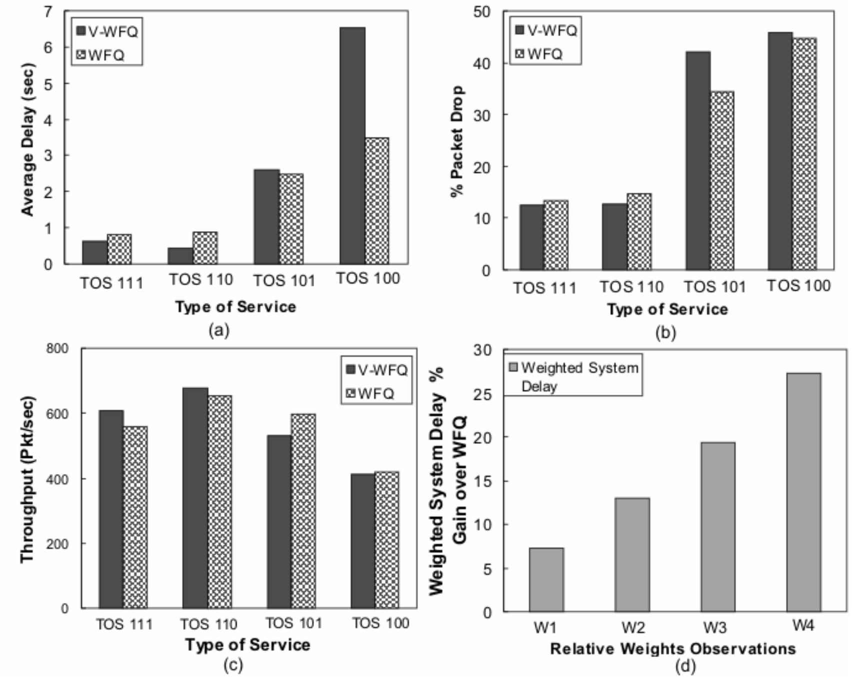

Figure 3: Experiment performance comparison results.

The performance results illustrate that V-WFQ performs exactly as intended compared to WFQ in terms of the end-to-end packet delay (i.e., better performance for high priority packets and poorer end-to-end delay for low priority packets). The unexpected results were that V-WFQ performed the same or better compared to WFQ in terms of throughput and packet drop rate for all packets. As explained previously, this was because of the use of additional queues in V-WFQ. It is also clearly shown that if one attaches weights (or values) to packets according to their priority level, V-WFQ achieves a better overall system value (based on the WASD metric) as compared to WFQ.

5. Conclusions

The proposed V-WFQ algorithm dynamically associates relative priorities for different types of services at run time based on the current congestion level at each individual router. V-WFQ requires more queues than WFQ in order to efficiently implement the switching among the queues based on the congestion level of the network. In our simulation experiments, the optimum number of queues for a 4 ToS level system was determined to be 6. Implementing V-WFQ with a lesser number of queues restricts the switching between queues; hence V-WFQ did not showed significant leverage over WFQ. Meanwhile, implementation of V-WFQ with more than 6 queues induced starvation of some flows as the relative difference between the priorities of queues became too significant. Ultimately, the determination of the optimum number of queues rests with the network system administrators based on the number of priority levels and experimental results. Our performance results show that when compared to WFQ, V-WFQ enhances the performance of higher priority traffic while providing comparable performance in terms of throughput and packet drop rate for low priority traffic.

References

1. A. Parekh, A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks, PhD thesis, MIT, Feb. (1992).
2. A. Demers, S. Keshav and S. Shenker, Analysis and Simulation of a Fair Queuing Algorithm, *In Proceedings of SIGCOMM ’89*, pp. 1-12, Sept. (1989).
3. D. Verma et al., A Framework for Differentiated Services, IETF draft 2475, June (1998).