A Novel Algorithm for Resource Allocation for Heterogeneous Traffic

Dequan Liu, Nirwan Ansari and Edwin Hou*
Advanced Networking Laboratory
Department of Electrical and Computer Engineering
New Jersey Institute of Technology, Newark, NJ
Email: {dxl021, ansari, hou}@njit.edu

Abstract-**To support different QoS requirements of heterogeneous and bursty traffic, Weighted Minmax algorithm (WMinmax) is proposed in this paper to efficiently and dynamically allocate network resources. In WMinmax, heterogeneous and regulated traffic are grouped into several classes according to their negotiated QoS parameters. For different classes, resources are allocated in proportion to their corresponding weights.**

Index Terms: *Resource allocation, asynchronous transfer mode, fairness, quality-of-service.*

I. INTRODUCTION

To efficiently allocate network resources to bursty traffic, many dynamic resource allocation algorithms have been proposed [1,2,3]. The allocated bandwidth for each flow is in proportion to its queue length in the Buffer–Population-Based Dynamic Slot Assignment algorithm [1] and the Generalized Longest Queue First (GLQF) algorithm [2]. In the Proportional Linear algorithm [3], in addition to the buffer occupancy, the instantaneous arrival rate is also included to calculate the bandwidth sharing of each flow. To allocate more bandwidth to the flow with larger queue length, the proportional polynomial algorithm [3] was proposed, where the bandwidth allocation is in proportion to polynomials of the sum of the queue length and instantaneous arrival rate. To achieve the desired goal of fair long–term buffer occupancy, the Minmax algorithm [3] was proposed by minimizing the maximum queue length of all the contending flows. The proportional exponential algorithm [3] was introduced to reduce the computational complexity of using Minmax algorithm. It was shown that the Minmax algorithm has the best performance among these algorithms in terms of cell delay, delay jitter and cell loss rate [3]. However, the Minmax algorithm is limited to handling homogeneous traffic only. To allocate resources for heterogeneous traffic that are classified according to their negotiated QoS parameters, we generalize the Minmax algorithm and propose the Weighted Minmax (WMinmax) algorithm.

This paper is organized as follows: The novel resource allocation algorithm along with the analysis are introduced in section II. Simulation results are given in section III.

*Corresponding Author

II. WEIGHTED MINMAX ALGORITHM

A. The Weighted Minmax Algorithm

Although the proposed algorithm is not limited to Variable Bit Rate (VBR) traffic, we will use it as the traffic model to simplify the description of our algorithm. VBR traffic are characterized by three parameters: Peak Cell Rate (PCR), Sustainable Cell Rate (SCR) and Maximum Burst Size (MBS) [4].

Following the model in [3], we assume N heterogeneous traffic streams share a link that is divided into time slots along the time axis. Each traffic stream is allocated a fixed bandwidth according to its SCR, and we consider the dynamic allocation of leftover bandwidth R to each stream based on its queue length and instantaneous rate. N traffic streams are classified into J classes according to their negotiated parameters and all traffic have identical parameters PCR, SCR, and MBS in each class. There are N_j streams in class j, $j=1,..,J$, and $\sum_{j=1}^{J} N_j = N$. In WMinmax, a weight is assigned to each class to make the bandwidth allocation relative to their negotiated parameters. Let L_i, λ_i and R_i be the queue length, arrival rate and allocated bandwidth of stream i, respectively, here $i=1,\ldots,N$. Also, let $Q_i=L_i+\lambda_i$, which is the estimated bandwidth needed to transmit the total traffic of stream i in next time slot. Then, the WMinmax algorithm is equivalent to solving the following optimization problem:

$$\text{Minimize } \{\max\{(Q_i - R_i)/w_i\}\} \qquad (1)$$

subject to:

$$\sum_{i=1}^{N} R_i = R$$

$$0 \le R_i \le Q_i$$

Here, w_i is the weight of stream i.

Note that WMinmax is a generalized version of the Minmax algorithm, i.e., the Minmax algorithm is a special case of WMinmax with all the weights being set to the same value. Like the Minmax algorithm, WMinmax assumes that all traffic conform to their negotiated parameters. Equation

(1) will be referred to as the normalization procedure. After the normalization procedure, WMinmax algorithm can be implemented the same way as the Minmax algorithm. It is clear that the larger the weight assigned to a stream, the more bandwidth will be allocated to that stream.

B. An Example

To illustrate the Wminmax algorithm, the scenario given in [3] will used as an example. There are two streams with rates $\lambda_1=1$ and $\lambda_2=2$ sharing the same output link with the rate $R=2$.

First we assume all weights are set to ones, and the initial queue lengths are zero. Since all weights are ones, we can apply the Minmax algorithm in [3] directly. At the first time slot, the expected queue lengths of the two streams are $Q_1=1$ and $Q_2=2$. So the allocated bandwidth for the two streams are $1/2$ and $3/2$, respectively. In the next time slot, actual queue lengths without including the estimated arrival rates are $L_1=0+1-1/2=1/2$ and $L_2=0+2-3/2=1/2$. In the following time slot, we have $L_1=1/2+1-1/2=1$ and $L_2=1/2+2-3/2=1$. Note that although the arrival rate of the second stream is two times the first one, from the above example, the allocated rate for the second stream is three times the first stream. In terms of equal allocation of bandwidth this is unfair. However, as argued in [3], from the perspective of the buffer content, the Minmax algorithm is fair for all streams.

Second, with the same scenario, we consider the case where the weights for the two streams are different: $w_1=1/3$ and $w_2=2/3$. Since the Minmax algorithm can only support homogeneous traffic, we have to use the WMinmax algorithm to achieve the different requirements for the heterogeneous traffic. In the first time slot, we will allocate $2/3$ and $4/3$ to the two streams, and we will have $L_1=0+1-2/3=1/3$ and $L_2=0+2-4/3=2/3$. In the next time slot, we have will allocate 2/3 and 4/3 to them, and have the queue lengths as $L_1=1/3+1-2/3=2/3$ and $L_2=2/3+2-4/3=4/3$. From the above example, we can conclude that a stream with a higher weight will obtain more bandwidth.

Comparing the above results of the two algorithms, it is clear that by introducing the weights to the corresponding streams, the WMinmax algorithm achieves better performance in terms of both bandwidth and buffer content for streams with different negotiated parameters.

C. Analysis of the Wminmax Algorithm

Let $S_i= L_i / w_i$, $i=1,\ldots,N$, where w_i is the corresponding weight of stream i, and $\sum_{i=1}^{m} w_i = 1$, $m \le N$, so S_i is the normalized queue length divided by its corresponding weight. Assume S_i be ordered as $S_1 \ge S_2 \ge \ldots \ge S_N$ and the WMinmax algorithm produces queue lengths of N traffic streams as: L_1, L_2, ..., L_N. Then we have can prove the following lemmas:

Lemma 1: There exists m and n, m≤n≤N, satisfying:

$$S_1 = \ldots = S_m = \ldots = S_n \ge S_{n+1} \ge \ldots \ge S_N$$

Proof:

Lemma 1 can be proved directly from the WMinmax algorithm, which allocates the available bandwidth to the first m largest streams, and makes their queue lengths equal with respect to their corresponding weights. Note that all the available bandwidth are completely allocated among the first m streams, and none are allocated to the remaining n-m streams which have equal queue lengths as the first m streams though.

■

Lemma 2: For any k≤n, we have:

$$S_k = (Q_1 + Q_2 + \ldots + Q_m - R)$$

Proof:

Based on the fact that all available bandwidth are completely allocated to the first m largest queues, and none to the remaining N-m queues, we have:

$$\sum_{i=1}^{m} S_i w_i = \sum_{i=1}^{m} Q_i - R$$

Since $\sum_{i=1}^{m} w_i = 1$ and $S_1 = \ldots = S_m$

We have:

$$S_1 = \ldots = S_m = (Q_1 + Q_2 + \ldots + Q_m - R)$$

Since all the largest n streams have the same queue lengths, we have

$$S_1 = \ldots = S_m = \ldots = S_n = (Q_1 + Q_2 + \ldots + Q_m - R)$$

■

Lemma 3: There exists m and n, m≤n≤N. such that for k=1,…,n. we have:

$$(Q_1 + Q_2 + \ldots + Q_m - R) < S_{k+1} \qquad \text{if } k<m \qquad (2)$$

$$(Q_1 + Q_2 + \ldots + Q_m - R) = S_{k+1} \qquad \text{if } m \le k<n \qquad (3)$$

$$(Q_1 + Q_2 + \ldots + Q_m - R) > S_{k+1} \qquad \text{if } k=n \qquad (4)$$

Proof:

For Equation (2), if there is a $k<m$ satisfying $(Q_1 + Q_2 + \ldots + Q_m - R) > Q_{k+1}$, then there is at least one stream $k=m-1$ will have no bandwidth allocated to it. This contradicts the results of the WMinmax algorithm. Therefore, Equation (2) holds for all k smaller than m.

Equations (3) and (4) can be proved directly from Lemma 2 and Lemma 1, respectively.

■

Lemma 3 provides us a way to implement the WMinmax algorithm.

Lemma 4: If the queue lengths of N traffic streams with zero initial queue lengths yielded by the WMinmax algorithm are: $(L,\ldots,L, L_{n+1},\ldots,L_N)$, $L>L_{n+1}$, then for any initial queue lengths in time slot j, we have:

$$\lim_{j\to\infty} \frac{L}{S_i^j} = w_i, \qquad if \quad i \le n$$

Proof:

First, we consider the condition that all the weights are set to one. Then divide the N streams into J subsets, and there are N_j streams in each subset with equal arriving rates. Here, $\sum_{j=1}^{J} N_j = N$. For different subsets, the corresponding arriving rates are different as: $\lambda_{j1}<\lambda_{j2}$, for $j1<j2$. Also, let L_i^j be the buffer content of the *ith* stream in subset j, we need to show that: for any subsets $j1$ and $j2$, $\forall i\in j1$ and $\forall k\in j2$ if $j1<j2$, then $Q_i^{j1} < Q_k^{j2}$.

The following four cases will happen at the current time slot:

Case 1, Bandwidths are allocated to both stream i and k. Then at the next time slot, the difference in queue length between the two streams remains the same as the previous time slot.

Case 2, Bandwidth is allocated to stream i and none to stream k (this case happens when $Q_i \geq Q_k$). The difference between the two is non-positive at the next time slot.

Case 3, Bandwidth are allocated to stream k and none to stream i (this case happens when $Q_i \leq Q_k$). The difference between Q_i and Q_j decreases at the next time slot.

Case 4, No bandwidth are allocated to both streams. Since $\lambda_i>\lambda_j$, the difference between Q_i and Q_j decreases at the next time slot.

Thus, Q_i will be no less than Q_j after several time slots.

Second, from lemma 3, the difference between the largest *m*th streams and the *(m+1)*th stream will be negative at the next time slot. Therefore, all queue lengths will eventually become equal after a period of time.

Let us assign weights w_i and w_j to stream i and j, respectively. To keep the ratio of their queue lengths to be w_i / w_j, the allocated bandwidth should be different for $w_i \neq w_j$. Thus, the stream with the same arriving rates but different weights will have different queue lengths respective to their corresponding weights.

This completes the proof of Lemma 4.

■

Since S_I is divided by its corresponding weight, it is the normalized queue length of stream i. From Lemma 4, we can conclude that the traffic with a larger weight will obtain more bandwidths than those with smaller weights so that the heterogeneous traffic can be efficiently supported by the WMinmax algorithm. The SCR of each stream can be selected as its corresponding weight.

III. SIMULATIONS

In our simulation, two ON-OFF traffic contend for the same out link which has a bandwidth *2M*. To find the worst case performance, we assume that the traffic models are extreme ON-OFF traffic [5].

In the first scenario, two streams have the same arriving traffic characters, i.e., both of them have the On time interval: *Ton=0.5* and Off time interval: *Toff=0.5*. So, the average load $\rho=0.5$. Peak rate *Rp=5M*, and their weights are assigned to *0.5* and *0.5*, respectively. In the second scenario, the two streams have the same On/Off time and the traffic load, but different peak rate: *Rp1=6.5M* and *Rp2=3.5M*, and their weights are assigned to *0.65* and *0.35*, respectively.

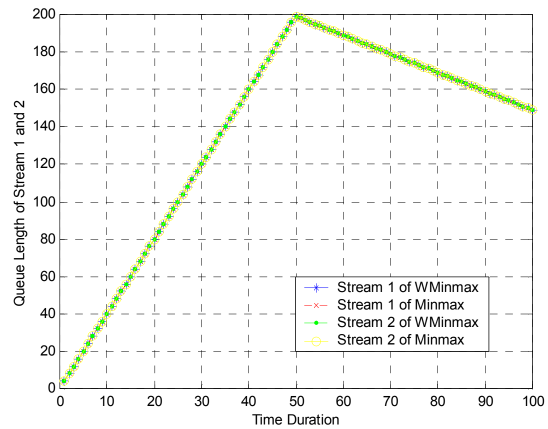

Fig. 1. Queue length of stream 1 and 2 via WMinmax and Minmax algorithm for the first scenario.

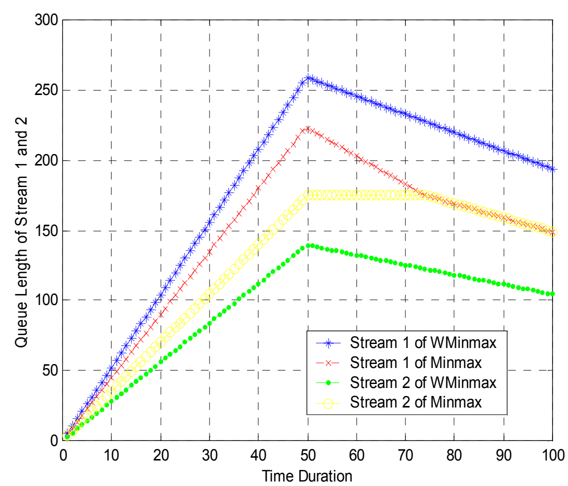

Fig. 2. Queue length of stream 1 and 2 via WMinmax and Minmax algorithm for the second scenario.

As shown in Fig. 1, if all traffic have the same traffic parameters and the same weights, the WMinmax and Minmax algorithms have the same performance. However, as shown in Fig. 2, by assigning different weights to the traffic with different traffic characteristics, the WMinmax algorithm can more efficiently allocate bandwidth among the contending traffic.

IV. CONCLUSIONS

To satisfy different QoS requirements of heterogeneous traffic, we propose the WMinmax algorithm in this paper. Each class of traffic is assigned with a weight, which reflects their respective bandwidth requirements. From the simulation, we can conclude that, by using the WMinmax algorithm, the different bandwidth requirements can be satisfied for the contending heterogeneous traffic.

ACKNOWLEDGMENTS

The authors would like to thank Surong Zeng and Jie Yang of NJIT for their discussions.

REFERENCES

[1] S. Chowdhury and K. Sohraby, "Alternative Bandwidth Algorithm for Packet Video in ATM Networks*," Proc. Infocom'92*, pp.1061-1068, 1992.

[2] D. Lee, 'Generalized Longest Queue First: An Adaptive Scheduling Discipline fro ATM Networks," *Proc. Infocom'97*, pp. 318-325, 1997.

[3] S. Biswas, and R. Izmailov,"Design of a Fair Bandwidth Allocation Policy for VBR Traffic in ATM Networks," *IEEE/ACM Trans. On Networking*, Vol. 8, No.2, pp. 212-223, Apr. 2000.

[4] Traffic Management Specification Version 4.1, af-tm-0121.000, *The ATM Forum*, March 1999.

[5] F. Presti, Z. Zhang, J. Kurose, and D. Towsley, "Source Time Scale and Optimal Buffer/Bandwidth Tradeoff for Heterogeneous regulated Traffic in a Network Node," *IEEE/ACM Trans. On Networking*, Vol. 7, No.4, pp. 490-501, Aug. 1999.