An Adaptive Scheduling Scheme for Fair Bandwidth Allocation

Wei Liu, Wenqing Cheng, Jianhua He, Chunhui Le, Zongkai Yang
Department of Electronics and Information Engineering
Huazhong University of Science and Technology, China
wliu@public.wh.hb.cn

ABSTRACT

Class-based service differentiation is provided in DiffServ networks. However, this differentiation will be disordered under dynamic traffic loads due to the fixed weighted scheduling. An adaptive weighted scheduling scheme is proposed in this paper to achieve fair bandwidth allocation among different service classes. In this scheme, the number of active flows and the subscribed bandwidth are estimated based on the measurement of local queue metrics, then the scheduling weights of each service class are adjusted for the per-flow fairness of excess bandwidth allocation. This adaptive scheme can be combined with any weighted scheduling algorithm. Simulation results show that, comparing with fixed weighted scheduling, it effectively improve the fairness of excess bandwidth allocation.

Keywords: Differentiated Services, Adaptive Weighted Scheduling, Bandwidth Allocation

1. INTRODUCTION

The Differentiated Services (DiffServ) architecture [1] is regarded as the most promising solution for the Internet Quality of Service (QoS) problem. Two Per Hop Behavior (PHB) groups, the Expedited Forwarding (EF) [2] and the Assured Forwarding (AF) [3], are specified beyond Best-Effort (BE) service in DiffServ architecture. EF service is proposed to provide a service with low loss rate, low delay and an assured throughput. AF service provides low loss rate and without assurance in delay and delay jitter, and it allows traffic flows to consume the remaining bandwidth in some fair manner under low traffic load.

In an interior DiffServ route, there are one EF, four AF and one BE service classes served at the output interface. The scheduling algorithm is responsible to adjust bandwidth among multiple service classes. The most widely deployed scheduling algorithms on DiffServ nodes are in the class of weighted scheduling, such as Weighted Round Robin (WRR) and Weight Fair Queueing (WFQ). In the weighted scheduling system, the queue of EF service should be assigned with highest priority and kept empty, while the queues of AF and BE service are serviced in priority or weighted fashion, and the excess (unsubscribed) bandwidth is equally shared among AF and BE flows.

There are two kinds fairness problems in sharing the bandwidth in DiffServ network: fairness within service class, or among service classes. The former fairness origins in the heterogeneity in traffic, such as the TCP flows with different RTT and UDP flows. The method to solve this fairness problem is to identify the misbehavior flows, such as by marking at the network edge [4, 5]. The latter fairness problem comes from the bursty characteristics of Internet traffic, such as the changes of active traffic flows or the subscription ratio of current link. In real networks, there are always excess bandwidth due to the overprovision network deployment. As to the DiffServ router adopted RIO (RED with IN/OUT) algorithm [6], the out-profile (OUT) traffic of AF services share the total unsubscribed bandwidth fairly with the traffic of BE service. If the bandwidth assigned to the queue of AF and BE services are fixed, the flows in AF services will get less excess bandwidth when the number of flows in BE service increases much. This is unfair and degrades the service differentiation. In this paper, We focus on scheduling based approach to solve the latter fairness problem in bandwidth allocation among service classes.

The basic idea of our solution is to dynamically adjust scheduling weights upon the changes of traffic load. There have been some related works following this idea [7–9]. The author in [7] hold the fairness criterion of

The work in this paper was supported by the National Natural Science Foundation of China (No.60202005).

Network Architectures, Management, and Applications II, edited by S. J. Ben Yoo, Gee-Kung Chang, Guangcheng Li, Kwok-wai Cheung, Proc. of SPIE Vol. 5626 (SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.574964

weighted differentiation, and proposed to adjusted the weights in proportion to the original weights of different service classes. The author in [8] and [9] aimed to achieve per-flow fairly sharing excess bandwidth, but they use different methods to estimate the number of active flows. [8] measured the arrival rate and used the Kalman filter estimation to get the number of flows. [9] adopted the method of *Zombie List* to estimate the number of active flows. However, the method in [8] was not compatible to DiffServ architecture since it marked the arriving rate in packet's header, while the method in [9] was not scalable since it required additional buffer to realize Zombie List. In this paper, we also follow the per-flow even-sharing fairness criterion. We propose a measurement-based approach to estimate the number of active flows and result in a new adaptive weighted scheduling scheme. This scheme can be combined with any weighted scheduling algorithm. Since the calculation of ideal weights is based on the measurement of local queue metrics, our scheme is more easier and scalable than the approaches in [8,9].

The rest of paper is organized as follows. In Section 2, the research scenario of scheduling system is introduced. Then Section 3 describe the estimation of subscribed bandwidth and the number of flows, and propose the adaptive weighting scheme. In Section 4, we evaluate our scheme with the original weighted scheduling algorithms in simulation. Finally, we conclude the paper in Section 5.

2. BACKGROUND

2.1. Research Scenario

A scheduling system model of multiple service classes in a DiffServ router is illustrated in Fig.1. Since we focus on the fairness problem of excess bandwidth allocation and EF service provide exact throughput guarantees, only the queues of AF and BE service classes are plotted in the figure.

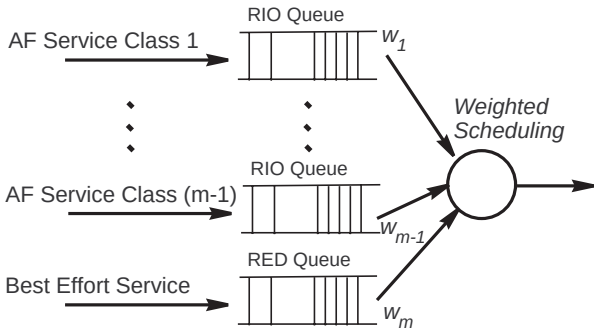


Figure 1. The scheduling system model with multiple service classes in DiffServ router

As specified in RFC2597 [3], there are at most four AF service classes in DiffServ node. In Fig.1, each service class is assigned a buffer queue. The four queues in AF service are served by RIO algorithm [6], and the queue in BE service is served by RED algorithm [10]. Weighted scheduling algorithm (such as WRR, WFQ, WF^2Q) is adopted at the output link interface. Since the flows in BE service have the same priority with AF flows in the competition for excess bandwidth, it can be regarded as a special AF service which has zero bandwidth subscription. We can use a group of AF services $i(i = 1, .., m,\ m \leq 5)$ to indicate the services queues in the scheduling system. The scheduling weights of each service class at the output link are represented as w_i in Fig.1.

To simplify our discussion, we make some assumptions as following: The total number of service classes m is fixed; the total bandwidth assigned to the AF and BE services B(in packets/sec) is also fixed; only traffic flows in TCP protocol are considered in this model, with the deployment of TFRC (TCP-friendly rate control) [11], our work can also be extended to UDP traffic as well.

2.2. Fairness criterion in excess bandwidth allocation

In this paper, we assume the network is configured under-subscribed condition, which means there is always excess bandwidth unsubscribed. The fairness criterion adopted is that each flow in AF and BE service should

have equal share of excess bandwidth. For the scheduling system illustrated in Fig.1, S_i denotes the subscribed bandwidth in ith service class; B_i denotes the assigned bandwidth to ith service class; $\hat{B}_i$ denotes the fair bandwidth allocation to ith service class; N_i denotes the number of active flows in ith service class. Then, the fairness criterion can be expressed as:

$$\frac{\hat{B}_i - S_i}{\hat{B}_j - S_j} = \frac{N_i}{N_j} \tag{1}$$

from (1), we can get:

$$\frac{\hat{B}_i - S_i}{B - \sum_{j=1}^{m} S_j} = \frac{N_i}{\sum_{j=1}^{m} N_j} \tag{2}$$

Then, we have the fair share for the ith service class:

$$\hat{B}_i = S_i + \frac{N_i}{\sum_{j=1}^{m} N_j} \cdot (B - \sum_{j=1}^{m} S_j) \tag{3}$$

As shown in (3), in order to calculate $\hat{B}_i$, the number of flows N_j and the subscribed bandwidth S_j in each service class j $(j = 1, 2, ..., m)$ are required. In next section, we will introduce new methods to estimate the two metrics, and propose an adaptive weighted scheduling scheme.

3. ADAPTIVE WEIGHED SCHEDULING SCHEME

3.1. Estimation of the number of flows

At DiffServ network edge, a packet is marked as IN if it is within the subscribed bandwidth, otherwise it is marked as OUT. Different TCP traffic flows in the same service class will be aggregated into TCP aggregates. For ith service in scheduling system in Fig.1, the aggregated throughput B_i with N_i flows in RIO queue can be given by the throughput formula of TCP aggregate in [12]:

$$B_i = \frac{3}{4} S_i + \frac{3k}{4} \cdot \sum_{r=1}^{N_i} \frac{1}{RTT_i} \sqrt{\frac{2}{p_{i,r}^{out}}} \tag{4}$$

where k denotes the average TCP packet size, $RTT_{i,r}$ and $p_{i,r}^{out}$ denote the average round trip time and average loss ratio of OUT packets respectively for the rth flows in ith service queue.

Since we focus on the fairness problem among service classes rather than that within one service class, we can assume all the TCP flows in ith service queue are homogeneous. To simplify our discussion, we assume all the TCP flows have the same round trip time as RTT_i and the same loss ratio as p_i^{out}. Then (4) can be rewrote as:

$$B_i = \frac{3}{4} S_i + \frac{3kN_i}{4RTT_i} \sqrt{\frac{2}{p_i^{out}}} \tag{5}$$

The loss ratio of OUT packets and that of whole RIO queue has following relationship:

$$p_i = p_i^{in} \cdot p_{mark} + p_i^{out} \cdot (1 - p_{mark}) \tag{6}$$

where p_i and p_i^{in} denote the loss ratio for whole RIO queue and that for IN packets respectively; p_{mark} denotes the marking probability at the network edge.

In the under-subscribed case, there will be no loss for IN packets, i.e. $p_i^{in} = 0$. In the ideal marking case, the p_{mark} can be represented by $p_{mark} = S_i/B_i$. Then (6) can be rewrote as:

$$p_i = p_i^{out}(1 - S_i/B_i) \tag{7}$$

On the other hand, the round trip time RTT_i consists of the link propagation delay (T_i) and the queuing process delay $\bar{Q}_i/B_i$, where $\bar{Q}_i$ is the average queue length of the RIO queue in ith service class:

$$RTT_i = T_i + \frac{\bar{Q}_i}{B_i} \tag{8}$$

Combining the above equations (5,7,8), we can get the expression on the number of flows in ith service class by the measurable local metrics of $\bar{Q}_i$, S_i and B_i:

$$\begin{aligned} N_i &= \frac{\sqrt{p_i}(B_i \cdot T + \bar{Q}_i)(4 - 3S_i/B_i)}{3k\sqrt{2(1 - S_i/B_i)}} \\ &\approx 2\sqrt{2p_i(1 - S_i/B_i)}(B_i \cdot T + \bar{Q}_i)/3k \end{aligned} \tag{9}$$

3.2. Estimation of subscribed bandwidth

In the under-subscribed case, traffic flows of AF service class can achieve their subscribed throughput in RIO queue. Since TCP flows are elastic traffic, the occupied queue length is non-zero at most time. Then we can get the following proportional relationship between the throughput and queue length at the steady state of RIO queue:

$$\frac{S_i}{B_i} \approx \frac{\bar{Q}_i^{in}}{\bar{Q}_i} \tag{10}$$

where $\bar{Q}_i^{in}$ is the average queue size of IN packets in RIO queue.

Hence, subscribed bandwidth S_i can be expressed as following approximately:

$$S_i \approx \frac{\bar{Q}_i^{in}}{\bar{Q}_i} \cdot B_i \tag{11}$$

3.3. Fair weights for excess bandwidth allocation

We investigate the relationship between current scheduling weights and ideal fair weights. Supposing w_i denotes the normalized current weight of service i, $\hat{w}_i$ denotes the normalized ideal fair weight of service i, then we can get:

$$\hat{w}_i = \frac{\hat{B}_i}{B} = \frac{S_i}{B} + \frac{N_i}{\sum_{j=1}^{m} N_j} \cdot (1 - \sum_{j=1}^{m} \frac{S_j}{B}) \tag{12}$$

where N_i and S_i $(i = 1, 2, ..., m)$ can be estimated by (9) and (11).

On the other hand, the outgoing bandwidth of each service class is proportioned to its scheduling weight in ideal weighted scheduling. Therefore:

$$B_i = w_i \cdot B \tag{13}$$

Supposing the total bandwidth of the link B and the propagation delay T known, by combining (9),(11),(13) with (12), we can get a function $f(\cdot)$ to calculate $\hat{w}_i$:

$$\begin{aligned} \hat{w}_i &= f(\bar{Q}_i^{in}, \bar{Q}_i, \bar{p}_i, \{w_j\}) \\ &\approx \frac{\bar{Q}_i^{in}}{\bar{Q}_i} w_i + \frac{\sqrt{\bar{p}_i(1-\frac{\bar{Q}_i^{in}}{\bar{Q}_j})}(w_i BT + \bar{Q}_i)}{\sum_{j=1}^{m} \sqrt{\bar{p}_j(1-\frac{\bar{Q}_i^{in}}{\bar{Q}_i})}(w_j BT + \bar{Q}_j)} (1 - \sum_{j=1}^{m} (\frac{\bar{Q}_j^{in}}{\bar{Q}_j} w_j)) \end{aligned} \tag{14}$$

3.4. Adaptive scheduling scheme

A periodical adaptive weighted scheduling scheme can be proposed based on function $f(\cdot)$: in every adaption period, the ideal weights $\{\hat{w}_i\}$ are calculated based on current scheduling weights $\{w_i\}$ as well as some measurable metrics, and then assigned to the actual scheduling weights.

It is obvious that our proposed adaptive scheduling scheme does not rely on the specific scheduling algorithm. The adaptive scheduling scheme can be combined with any weighted scheduling algorithm. The complete process of the combined scheduling are described as follows.

1. At the beginning of every adaption period τ, all the variables are reset.
2. If one packet in class i dropped, the drop counter D increased by 1.
3. At every sampling interval τ_s, the current queue length $\{Q_{j,k}\}(k = 1, .., \tau/\tau_s)$ and that of IN packets queue $\{Q_{j,k}^{in}\}$ in each service class j $(j = 1, .., m)$ are recorded.
4. At the end of the period τ, the average dropping probability $\bar{p}_j$, the average queue length of whole queue $\bar{Q}_i = \tau_s/\tau \cdot \sum Q_{j,k}$ and the average queue length of IN packet $\bar{Q}_i{}^{in} = \tau_s/\tau \cdot \sum Q_{j,k}^{in}$ in each service class j are calculated.
5. With (9) and (11), the number of active flows N_j and the subscribed bandwidth S_j in each service class are estimated, and the normalized weights for next period $\{\hat{w}_j\}$ are calculated according to (12).
6. The adaptive scheduling scheme adjusts the bandwidth for each service according to the weights $\{\hat{w}_j\}$.

3.5. Discussion

This adaptive scheduling scheme are based on the metrics estimated or measured from local queue metrics, such as $\bar{Q}_{j,k}$, $\bar{Q}_{j,k}^{in}$ and p_i. The other metrics used in our scheme can also be easily obtained. For example, there are known methods estimating the propagation delay T [5]. Comparing with the method in [8] and [9], our scheme is more scalable and easier to deploy.

The length of adaptation period in our scheduling scheme should be carefully determined. If the period is too short, the frequency of adaptation will be larger which results in unnecessary system load. If the period is too long, it will be difficult to capture and respond to the bursty traffic. In real networks, it is suggested to consider the characteristics of bursty traffic in local node. In our simulation, the period is set as 5 to 10 seconds.

This scheme should be deployed as RIO algorithm in queue management. The estimation on the number of flows in this scheme is based on average queue length. Since true condition of (11) is that the size of queue buffer is large enough and the occupancy is steady, the estimation will be very inaccurate if the queue size is too small. In the deployment of our scheme, we suggest that: (1) the size of queue buffer is configured to integer times of the bandwidth-delay product of this link; (2) the maximal queue threshold for BE service or the OUT packets in AF service are set around 0.7 times of the buffer size.

4. SIMULATION RESULTS

The proposed adaptive weighting scheme is implemented in ns-2 [13] and evaluated by two groups of simulations. The network topology in simulation is shown in Figure.2. It illustrates a simple DiffServ domain with two edge routers (*E0,E1*) and one core routers (*C0*). TCP connections are setup from the source nodes (*S0,S1*) to the destination nodes (*D0*). The bottleneck link in DiffServ Domain is assigned with 50Mbps bandwidth and 10ms propagation delay, while all the access link are 100Mbps bandwidth and 10ms propagation delay. Hence, the total propagation delay in a round trip is in 80ms.

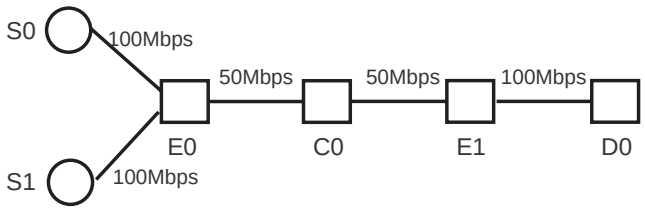


Figure 2. The topology of the simulation

All the TCP packets from *S1* to *D0* are assigned in BE service, while those from *S0* to *D0* are assigned in AF service. The RED queue in BE service on *C0* are configured as $(th_{min}, th_{max}, p_{max}) = (80, 200, 0.02)$. The RIO queues in AF services are configured as $(th^{out}_{min}, th^{out}_{max}, p^{out}_{max}) = (50, 150, 0.1)$ and $(th^{in}_{min}, th^{in}_{max}, p^{in}_{max}) = (70, 200, 0.02)$ for OUT and IN part RED respectively. The marking algorithm deployed at network edge is Token Bucket marking. We deployed original WRR scheduling algorithm on *C0* as well as our proposed scheduling scheme. We name the combination algorithm of original WRR with adaptive scheduling scheme as adaptive WRR (AWRR) in this paper. The adaptation period is 10 second in simulation.

In order to evaluate the fairness allocation of excess bandwidth among of all the flows in AF and BE service classes, we defined a fairness index as below:

$$Fairness \;\; Index = \frac{(\sum_{i=1}^{m} e_i)^2}{m \cdot \sum_{i=1}^{m} (e_i{}^2)} \tag{15}$$

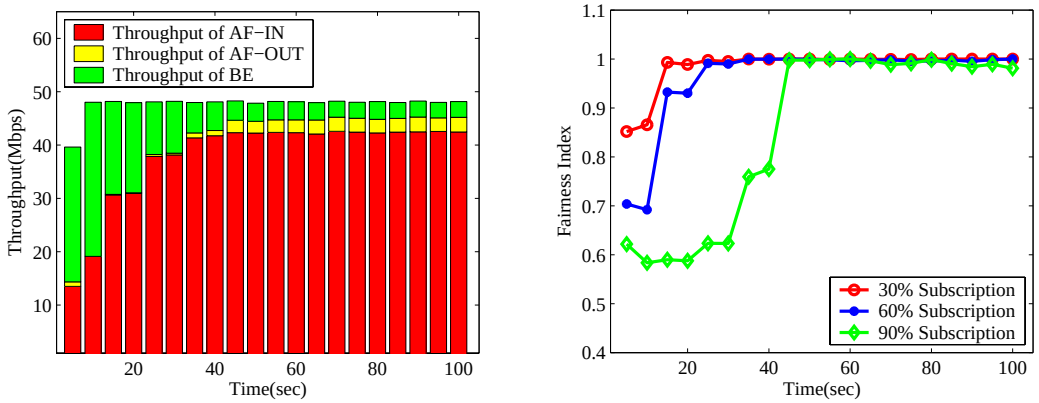
where n is the total number of flows in AF and BE services, e_i is the measured excess bandwidth of flow i. It is obvious that the fairness index of 1 indicate the ideal fair case.

4.1. The effect of subscription ratio

In DiffServ networks, subscription ratio is an important factor in traffic load. As indicated in (3), there is a tight relationship between the fair assigned bandwidth of each service queue and its subscription bandwidth. In real networks, the subscription ratio may be re-configured by ISP. In the first group of simulation, we investigate the ability of our proposed scheme responding to various conditions of subscription ratio.

We keep the number of flows fixed in the simulation, set the number of flows of AF and BE service as 25 and 35 respectively and set the initial scheduling weights between AF and BE services as (40:60). We run AWRR algorithm several times, and the subscription ratio in each time changed from 30% to 90%. The IN and OUT throughput of AF service and that of BE service are recorded. The fairness index of excess bandwidth allocation is calculated. Simulation results are shown in Fig.3.

Due to space limited, only the bar picture of bandwidth allocation among AF-IN , AF-OUT and BE service under 90% subscription is given in Fig.3(a). We can observe that AWRR achieves stable fair bandwidth allocation at 50 second. When the time at 5 second, the bandwidth allocation is still determined by the initial weights setting (AF:BE=40:60); after 5 times of adaptation and time at 50 second, the allocation is almost near ideal weights



(a) Bandwidth allocation: AWRR, 90% subscription

(b) Fairness index for allocation of excess bandwidth

Figure 3. Adaptive scheduling under different subscription ratio

as (AF-IN:AF-OUT:BE=90:4:6). Fig.3(b) shows the fairness index of AWRR at different time under various subscription ratio. We can find that, in most cases, AWRR can quickly adjust the fairness of excess bandwidth to ideal value 1. Thus, the adaptive scheduling scheme has the ability to achieve fair excess bandwidth allocation regardless of the original settings of original weights.

4.2. The effect of dynamic traffic load

In DiffServ networks, the number of flows is another important factor in traffic load. In real networks, the traffic load of BE service is unknown and may changes over time. In the second group of simulation, we investigate the ability of proposed scheme responding to the changes of BE traffic load.

We keep the subscription ratio fixed as 60% in simulation, set the number of AF flows as 20. The number of BE flows changes over time, as shown in the Fig.4. The initial scheduling weights of AF and BE service are set as (50:50). We run simulations with AWRR and original WRR respectively. The the IN and OUT throughput of AF and that of BE service are recorded, and the fairness index for excess bandwidth allocation is calculated.

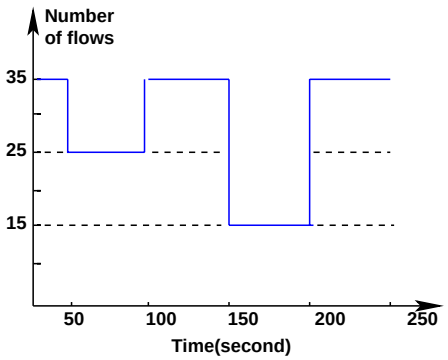
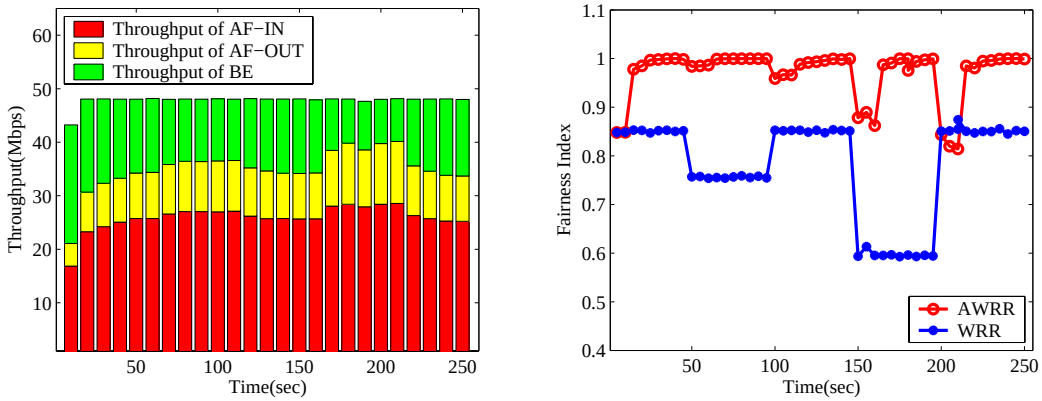


Figure 4. Dynamic traffic load in BE service

From Fig.5(a), we can observe the ability of proposed scheme responding to the changes of traffic load. Whatever the changes of traffic load, AWRR can quickly achieve the fair bandwidth allocation. For example, when time at 120 second, the number of BE service is 35, the bandwidth allocation scheduled by AWRR is set near to the ideal value of (AF-IN:AF-OUT:BE=60:15:25); when time at 180 second, the number of BE service is



(a) Bandwidth allocation: AWRR, 60% subscription

(b) Fairness index for allocation of excess bandwidth

Figure 5. Adaptive scheduling under dynamic traffic load

15, the output allocation scheduled by AWRR is set near to the ideal value of (AF-IN:AF-OUT:BE=60:23:17). On the other hand, the steps of adaptive scheduling scheme over large changes of traffic load can also be observed in Fig.5(b). When time at 150 second and 200 second, AWRR spend two more adaption times to adjust the fairness index near to ideal 1. In most of the time, AWRR can maintain the near 1 fairness index for excess bandwidth.

5. CONCLUSIONS

In this paper, an adaptive weighted scheduling scheme is proposed for DiffServ router to achieve the fair allocation of excess bandwidth among the flows in AF and BE services. Measurement-based approaches are proposed to obtain the estimations of the number of traffic flows and subscribed bandwidth. Simulations are designed to evaluate the ability of the scheme responding to the changes of traffic load. The results show that the scheme is robust under different subscription situations. It can not only achieve ideal weights allocation quickly, but also respond to the dynamic changes of traffic load. Compared with fixed weighted scheduling, this scheme can achieve effectively improve the fairness of excess bandwidth allocation in DiffServ routers.

REFERENCES

1. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for Differentiated Services." RFC 2475, Dec. 1998.
2. B. Davie, A. Charny, J. Bennett, K. Benson, J. L. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)." RFC 3246, Mar. 2002.
3. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group." RFC 2597, June 1999.
4. H. Su and M. Atiquzzaman, "ItswTCM: a new aggregate marker to improve fairness in diffserv," *Computer Communications.* , pp. 1018–1027, 2003.
5. M. El-Gendy and K. Shin, "Assured forwarding fairness using equation-based packet marking and packet separation," *Computer Networks.* , pp. 435–450, 2003.
6. D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Trans. on Networking.* **6**, pp. 362–373, Aug. 1998.
7. L.Ji, T. Arvanitis, and S. Wolley, "Fair weighted round robin scheduling scheme for DiffServ networks," *Electronics Letters.* , pp. 333–335, Feb. 2003.

8. R.Kawahara and N.Komatsu, "Dynamically weighted queueing for fair bandwidth allocation and its performance analysis," in *Proc. of IEEE ICC'02.*, pp. 2379–2383, June 2002.
9. H.Shimonishi, I.Maki, T.Murase, and M.Murata, "Dynamic fair bandwidth allocation for diffserv classes," in *Proc. of IEEE ICC'02.*, pp. 2348–2352, June 2002.
10. S.Floyd and V.Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. on Networking.* **1**, pp. 397–413, Aug. 1993.
11. M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol specification." RFC 3448, Jan. 2003.
12. I. Yeom and A. Reddy, "Modeling TCP behavior in a Differentiated Services network," *IEEE/ACM Trans. on Networking.* **9**, pp. 31–46, Feb. 2001.
13. Information Sciences Institute in University of Southern California, "The Network Simulator(ns-2)." http://www.isi.edu/nsnam/ns.