Adaptive Scheduling using Online Measurements for Efficient Delivery of Quality of Service

Peng Xu, Michael Devetsikiotis and George Michailidis

Technical Report #2004-12
March 29, 2004

This material was based upon work supported by the National Science Foundation under Agreement No. DMS-0112069. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Statistical and Applied Mathematical Sciences Institute
PO Box 14006
Research Triangle Park, NC 27709-4006
www.samsi.info

Adaptive Scheduling using Online Measurements for Efficient Delivery of Quality of Service

Peng Xu*, Michael Devetsikiotis* and George Michailidis**

*Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC

**Department of Statistics
University of Michigan, Ann Arbor, MI

Abstract

In this paper we introduce and investigate the properties of a new dynamic resource allocation scheme that utilizes online measurements to achieve the required QoS. The proposed scheme can be implemented under various differentiated network frameworks, such as IntServ, DiffServ, and Traffic Engineering with MPLS. The new scheme adjusts the weights of the underlying packet scheduling algorithm, such as weighted round robin or weighted fair queueing, by utilizing the information contained in traffic measurements obtained over successive time windows. Extensive experimental evidence suggests that the proposed scheme exhibits an improved QoS performance for variable traffic loads over classical scheduling algorithms. In addition, the scheme proves robust to the choice of several of its key tuning parameters.

I. Introduction and Overview

A. Motivation

The advent of bandwidth and delay-sensitive applications such as voice over IP (VoIP), video-conferencing, online gaming, interactive television, etc., has led to a requirement for scheduling algorithms that provide Quality of Service (QoS) guarantees to the application users.

Whether on a per-flow basis as in IntServ or on a per-class basis as in DiffServ, resources need to be reserved and packets need to be appropriately forwarded [1], [2], [3]. QoS can also be delivered by using label-based forwarding (e.g., MPLS) and traffic engineering principles [4]. The Traffic Engineering approach in the MPLS model uses a dynamic bandwidth allocation model, providing the bandwidth to a connection just for the time it is actually requested, thus improving the flexibility of the network [4], [5].

Any QoS delivery architecture has to rely on schedulers capable of differentiating among traffic classes. However, the inherent high variability of the traffic characteristics generated by such applications implies that *static* bandwidth reservation protocols accompanied by over-provisioning of network links leads to a significant under-utilization of available resources. On the other hand, a *dynamic* allocation of bandwidth that closely tracks the prevailing traffic characteristics could achieve significant savings, while at the same time satisfying QoS provisions (e.g., end-to-end delay, jitter or packet loss probability), as guaranteed by service-level agreements (SLAs). Implementation of such dynamic schemes requires an efficient traffic monitoring and estimation mechanism followed by an adaptive bandwidth allocation policy.

B. Overview

Significant contributions have been made already in related areas such as measurement-based admission control (MBAC), self-sizing network frameworks and QoS adaptive routing.

Traffic measurement and estimation has been widely studied in the past decade due to the importance of its accuracy and efficiency on QoS performance. *Effective Bandwidth* (EB) [6] is a well-known concept in this field that aims to allocate an efficient amount of bandwidth in order to satisfy QoS requirements of the incoming traffic. In this paper, we denote the effective bandwidth by EB, and assume it is calculated from a measurement algorithm.

Approaches to QoS could be differentiated by the combination of traffic measurement with different components under control and data plane of QoS [7]. MBAC [8], [9], [10], [11] is one of the first approaches to utilize the traffic measurement and estimation in order to make admission decisions based on QoS requirements.

In regard to traffic engineering, a QoS routing mechanism based on global control was proposed in [12], [13]. Alternatively, self-sizing frameworks in which online measurements are utilized for optimization and QoS routing (under global control) have been proposed in [14], [15]. Recently, due to the large overhead involved in signaling for global control, Z. Zhang *et al.* proposed QoS routing based on local control [16]. And the counterpart of self-sizing framework based on local control is studied by Nalatwad and Devetsikiotis in [17].

In QoS scheduling, Shin *et al.* have proposed the adaptive allocation of weights according to the average queue length of the premium service, in which only QoS constraints of premium service are considered [18]. Most recently, Chandra *et al* [19] describe a dynamic resource allocation technique that uses on-line measurements. However, other than [19], there have been limited advances in formally defined, control-theoretic closed-loop methodologies.

In this paper, we are extending these approaches by describing in detail an adaptation mechanism for generalized schedulers under periodic estimates of traffic and system state. Furthermore, we conduct extensive experiments investigating the effect that various factors have in the performance and robustness of such schemes. Finally, we initiate a formal description of this scheme by sketching the analytical derivation of the optimal settings.

C. Roadmap

Regardless of the specific QoS delivery mechanism, present or future, there are great efficiency and robustness advantages to be gained from enhanced, measurement-based algorithms for adaptation of scheduler settings. Our adaptive technique for generalized schedulers and its analysis presented here, apply equally well to any of these QoS mechanisms. In this paper, we introduce a measurement based QoS scheduling scheme implemented within a generalized (GPS, WFQ, or WRR) scheduling framework.

In the proposed approach we combine aspects of traffic measurement and estimation with scheduling algorithms and examine their performance extensively via simulation. We also outline the derivation of optimal control parameters, the formal part of which constitutes our on-going work. We extend and generalize previous approaches by proposing and formally describing a measurement-based QoS scheduling algorithm which aims to maximize the fairness of resource allocation among all classes of traffic, while satisfying their own QoS requirements.

The paper is organized as follows: Section 2 introduces the proposed measurement-based scheduling scheme together with the resulting decision policy, while in Section 3 an extensive performance evaluation study of the proposed approach is undertaken and its robustness with respect to key parameters shown. Some concluding remarks are drawn in Section 4. Finally, we outline our derivation of optimal scheduler settings in the Appendix.

II. A Measurement-Based Adaptive QoS Scheme

A. Preliminaries

Without loss of generality, we assume that our proposed measurement-based adaptive QoS scheme operates under the DiffServ mechanism, where the various traffic flows can be classified as: Expedited Forwarding (EF) service, Assured Forwarding (AF) service and Best Effort (BE) service. At an appropriate level of abstraction, these three service classes can be thought of as delay-sensitive, loss-sensitive and best effort, respectively, a characterization that we adopt for the remainder of the paper.

We focus on WFQ and WRR-like algorithms. The virtual time of WFQ is equivalent to its round number being calculated from a bit-by-bit round robin scheduler [20]. If a packet arrives in an inactive queue, then its virtual finish time is the sum of the recomputed virtual time (round number) and the service time for this packet. If the packet arrives in an *active* queue, then the virtual finish time is the sum of the last packet finish time plus the service time for this packet. This can be expressed as follows:

$$F_i^k \leftarrow \max\{V(a_i^k), F_i^{k-1}\} + L_i^k/\phi_i$$

where $V(a_i^k)$ is the virtual time of the kth packet of connection i, L_i^k is the length of the kth packet of connection i and ϕ_i is the service rate allocated to connection i. A connection is defined as active in WFQ if the last packet served from it, or in its queue, has a finish time greater than the current round number. Since WFQ needs to keep track of active connections to calculate its virtual time, the computation complexity of this algorithm is proportional to $O(N)$, where N is the number of active connections.

Moreover, the definition of WRR shows that the class weights correspond to the number of slots to be served in a single scheduling cycle. Hence, for both WRR and WFQ, the bandwidth allocated to class i at decision time τ is given by $\phi_i = w_i C / \sum_{j \in N(\tau)} w_j$, where w_i is the weight allocated to the i-th class and $N(\tau)$ is the number of backlogged queues at time instant τ.

The main components of our adaptive scheme are: a *traffic measurement module* that provides an accurate estimation of the future traffic load of the different classes under consideration, a *scheduling module* that deals with the packet forwarding mechanism and a *decision module* that determines how bandwidth is distributed among the various classes of traffic.

The coordination of these three components is described next: when a packet arrives at the scheduler, it is assigned to a certain traffic class and appended to the corresponding queue, waiting to receive service from the scheduling module. At the same time, the measurement module updates the arrival rate statistics of the corresponding traffic class, and provides an estimate of the *effective bandwidth*. It should be noted that the measurement module performs the above operation over a pre-specified time interval (window). Finally, the decision module allocates the weights to the queues using the EB information.

We elaborate next on the proposed decision policy and the measurement mechanism.

B. Decision Policy

Notice that transient phenomena, such as traffic burstiness, coupled with imbalances in the traffic load over classes, would result in performance degradations, unless the class capacity allocations are accordingly adjusted. A simple and intuitive rule that satisfies the QoS requirements is that at every point in time we have $\phi_i C \geq EB_i$, where ϕ_i is the share of the bandwidth of traffic class i, C the capacity of the output link and EB_i the estimated effective bandwidth for the i-th class. In practice, we require that the above relation holds for a time interval of length W, appropriately chosen (for more on the choice of W see Section III).

Nevertheless, in many instances, the sum of the estimated arrival rates could be greater than C, which corresponds to the under-provisioning case in the paper. In such instances, the QoS requirements of the delay-sensitive class should be satisfied at the expense of the remaining two classes, provided that it does not exceed a prespecified threshold θ_1. Furthermore, if the under-provisioning is due to the temporal traffic pattern of the best effort class, then the QoS requirement of the loss sensitive class should also be satisfied, provided that it does not exceed a different prespecified threshold θ_2. Summarizing the above requirements, the queue weights are determined as follows:

1) Obtain the effective bandwidth, EB_i for the i-th class from the measurement module.
2) If the sum of the three effective bandwidths is less than the link capacity (i.e., $\sum EB_i < C$), then each class' share of bandwidth is given by $\phi_i C \geq EB_i$; otherwise, go to step 3.
3) If $\sum EB_i > C$, then provide bandwidth to the delay-sensitive class up to $\phi_1 \leq \theta_1 C$, to the loss-sensitive class up to $\phi_2 \leq \theta_2 (EB_2/C)$ and the remainder to the best effort class.

4) If the scheduling module is based on WRR, then normalize the share of bandwidth for each class in order to get the integer weights allocated to them. If it is based on WFQ, then the share of bandwidth for each class corresponds to the weights.

We are currently formalizing the analysis of our algorithm by describing *analytically* the optimal choices for the bandwidth share for each class, ϕ_i. Due to space limitations and the scope of this experimental paper we only include an outline of our analytical methodology in the Appendix.

Finally, notice that the proposed approach also decreases the computational complexity of the resulting DWFQ scheme, since it is determined by the number of classes, as opposed to the number of connections.

C. Measurement Algorithm

The traffic envelope [11] has proved useful in obtaining online measurements. Furthermore, it turns out to be robust to the time dependence structure of traffic (e.g. Long-Range Dependence vs Short-Range Dependence). A brief description of the traffic envelope approach is given next.

Its basic measurement unit is the measurement slot, τ. A measurement window is adaptive, and comprised of varying number of measurement slots, $W_k = k\tau(k = 1, 2, \cdots, T)$. In a certain measurement window W_k, let $A[t, t+W_k]$ denote the counting process of arrivals in the interval $[t, t+W_k]$; thus, $A[t, t+W_k]/W_k$ is the arrival rate over that interval. The maximal rate for W_k over this time interval could be defined as $R_k = \max_t A[t, t+W_k]/W_k$.

Suppose $A_t = A[t\tau, (t + 1)\tau]$ are the arrivals in the time slot starting from t. In this way, the maximal rate over the certain measurement window with the size of $k\tau$, for the past $T\tau$ from the current time t could be obtained by

$$R_k^1 = \frac{1}{k\tau} \max_{t-T+k \le s \le t} \sum_{u=s-k+1}^{s} A_u \quad \text{for} \quad k = 1, 2, \cdots, T. \tag{1}$$

This equation is introduced for considering burstiness over small time scales.

The current envelope R_k^1 is measured and updated every $T \cdot \tau$ measurement window, $R_k^n \leftarrow R_k^{(n-1)}$ for $k = 1, 2, \cdots, T$ and $n = 2, 3, \cdots, N$. The variance between envelopes over the past N windows could be computed by the following equation:

$$\sigma_k^2 = \frac{1}{N-1} \sum_{n=1}^{N} (R_k^n - \bar{R}_k)^2 \tag{2}$$

where $\bar{R}_k = \frac{1}{N} \sum_{n=1}^{N} R_k^n$ is the mean of past N envelopes.

The effective bandwidth in traffic envelope can also be calculated in both the small and the large time scales [21]. For the large time scale, the effective bandwidth is obtained by

$$EB_{large} = \bar{R}_T + \alpha_{large}\sigma_T \tag{3}$$

where $\bar{R}_T$ and σ_T are the mean and deviation for past N envelopes with the measurement window size of $T \cdot \tau$. And α_{large} is used to specify the confidence interval. It can be computed by the inverse of complementary CDF of an N(0,1) Gaussian distribution, $\alpha_{large} = Q^{-1}(\frac{\epsilon \bar{R}_T}{\sigma_T})$.

For the small time scale, the EB is computed by

$$EB_{small} = \max_{k=1,2,\cdots,T} \frac{(\bar{R}_k + \alpha_{small}\sigma_k)k\tau}{k\tau - B/C} \tag{4}$$

where B and C are buffer size and capacity respectively. The mean $\bar{R}_k$ and deviation σ_k is for measurement window $k \cdot \tau$.And $\alpha_{small} = Q^{-1}(\frac{\epsilon \bar{R}_k}{\sigma_k})$ is computed by using the same approach as α_{large}.

The algorithm gives the worst case effective bandwidth by choosing the maximum between the small-scale effective bandwidth and the large-scale one:

$$EB = \max\{EB_{large}, EB_{small}\} \tag{5}$$

III. Performance Evaluation and Validation

A. Simulation Scenarios

The model used in our experimental investigations is based on the architecture shown in Figure 1. The network element is a processor sharing system with three input queues that map to the delay-sensitive, the loss-sensitive and best effort class, respectively. Each queue is associated with a certain weight whose value is controlled by

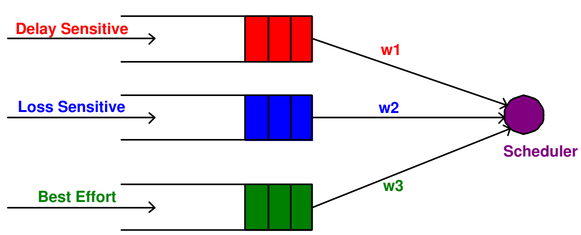

Fig. 1. The architecture of the simulation model.

the scheduling module. Our proposed scheduling scheme changes the weights *dynamically* over the adaptive time window, after their values are initially set at $w_1 = 2, w_2 = 2$, and $w_3 = 1$, reflecting the importance of the respective classes, for all scheduling algorithms in our simulation study. The link capacity of the scheduler is 64kbps and the buffer sizes for the three classes are set to 500 (delay-sensitive), 5000 (loss-sensitive), and 10000 (best effort) bytes, respectively.

The effective bandwidth is recalculated for every adaptive window $N \cdot T \cdot \tau$, with $\tau = 0.01$, $T = 10$, and $N = 10$. The queue management mechanism used corresponds to Drop-Tail.

The traffic generation model follows the on-off Fractal Modulated Poisson process (FMPP) proposed in [22]. The FMPP produces long-range dependent traffic, by using a power-law distribution for the “on” or “off” periods, while the packet distribution Over the "on" is Poisson. The traffic rate varies over time due to the changing number of flows in the "on" state. Furthermore, in our setting additional variability is introduced through changes in the composition of traffic over the three classes. For the sake of simplicity the packet size was fixed to 50 bytes and the Hurst parameter of the traffic process to 0.7.

Given the links speed, the system's capacity (stability) constraint is at 6.25 msec. That is, if the mean packet inter-arrival time is larger than this value, the system is over-provisioned and relatively few packet losses are expected to occur. On the other hand, for mean packet inter-arrival times smaller than 6.25 msec, the system is unstable and highly backlogged queues together with frequent packet losses are expected. The case where the mean time is exactly 6.25 corresponds to the critical regime.

Finally, the policies to be compared in the simulation study are the Static Weighted Round Robin (SWRR), the Static Weighted Fair Queue (SWFQ) and their dynamic counterparts DWRR and DWFQ. Their main difference is that for the static policies the set of weights do not change over time. The performance metrics used in our study are packet loss probabilities and average queue delays.

B. Results and Analysis

We start by providing a picture of the traffic patterns used in our study. Figure 2 shows how the total traffic rate and that of the three classes changes over time. Every 100 seconds, the traffic pattern of each class changes, which induces a transient bursty behavior for the traffic of each class. And Figures 3–5 demonstrate the traffic rate and allocated capacity by different scheduling policy schemes in under-provisioned case. We assume that the queues are always backlogged over each adaptive window.

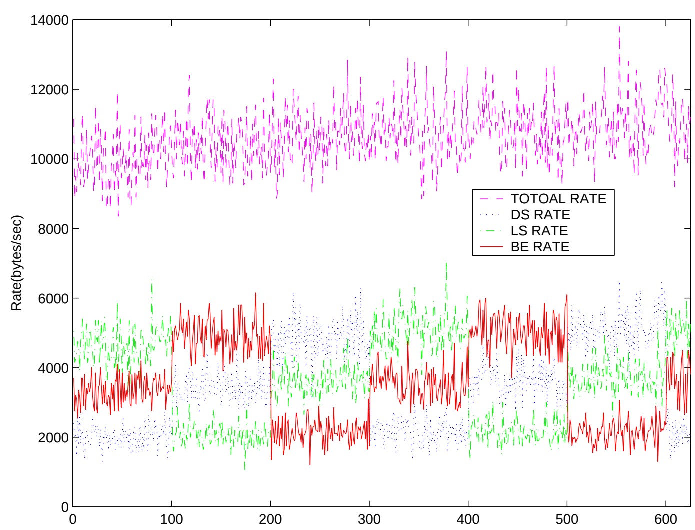

Fig. 2. Pictorial view of total traffic rate vs. traffic rate for each class in the under-provisioned case.

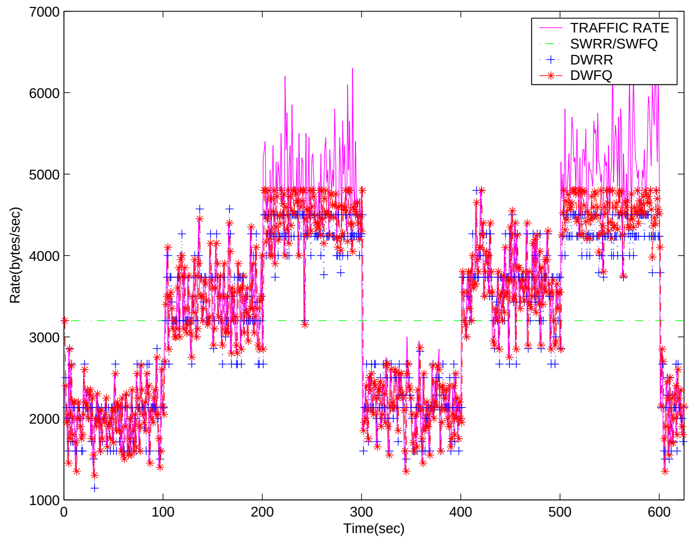

Fig. 3. Pictorial view of traffic rate vs. resource allocated by different scheduling algorithms for delay-sensitive class of traffic in the under-provisioned case.

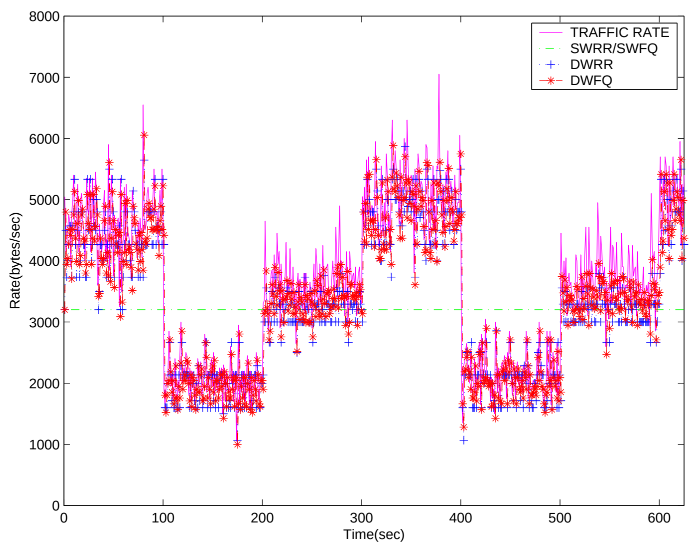

Fig. 4. Pictorial view of the traffic rate vs. resource allocated by different scheduling algorithms for the loss-sensitive class of traffic in the under-provisioned case.

The plots indicate that, in our simulation scenarios, the delay and loss sensitive classes should be adequately provisioned under the proposed QoS scheme. We turn our attention next to evaluating the performance of our scheme. In order to be able to examine a large range of performance the mean inter-arrival packet time is varied from 5 msec to 9.1 msec, thus covering both the under and the over-provisioned cases. In the following figures the class loss probabilities and average queue delays are plotted as functions of the input rate for the two dynamic and the two static policies.

Figures 6 (a) and 6 (b) show the substantial decrease in loss and delay obtained from the DWFQ policy for the delay sensitive class in the under-provisioned scenario. This policy also achieves the best performance in the well provisioned case, but the gains are smaller in the over-provisioned case because the bandwidth will be

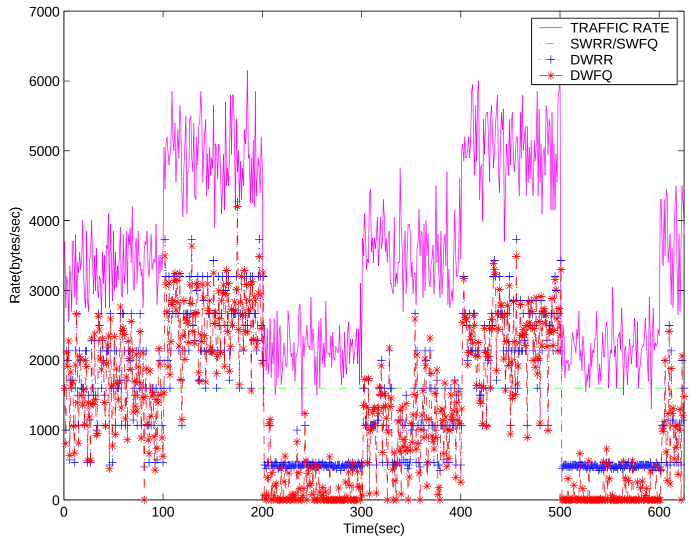

Fig. 5. Pictorial view of the traffic rate vs. resource allocated by different scheduling algorithms for best effort class of traffic in the under-provisioned case.

automatically redistributed among the rest of the nonempty queues whenever any queue becomes empty due to the work-conserving nature of the static scheduling policy. It is also worth noting that the DWRR policy outperforms both static policies under the loss performance metric for the same class, although this comes at the expense of larger delays. The latter is a consequence of the integer weight normalization for DWRR.

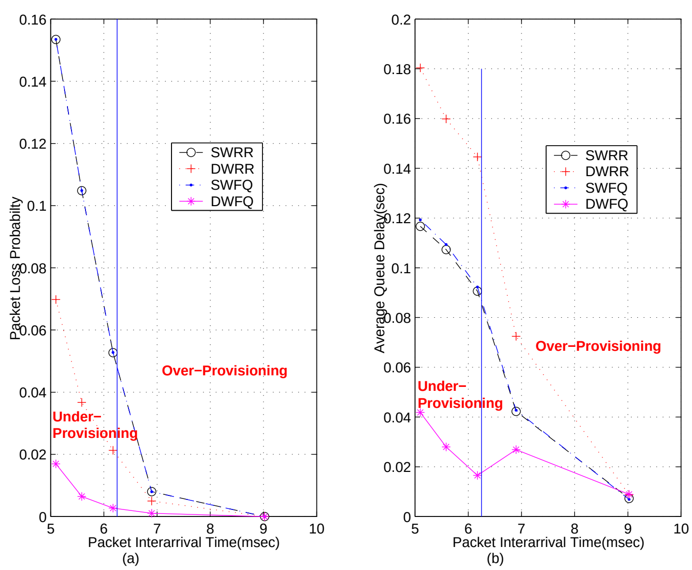

Fig. 6. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for delay-sensitive class.

Analogous conclusions can be reached for the loss-sensitive class by examining Figures 7 (a) and 7 (b). However, since queueing delays are not that important for this class, it can be concluded that DWRR outperforms the static policies, as well.

Finally, the dynamic policies prove competitive with respect to the best effort class, as Figures 8 (a) and 8 (b)

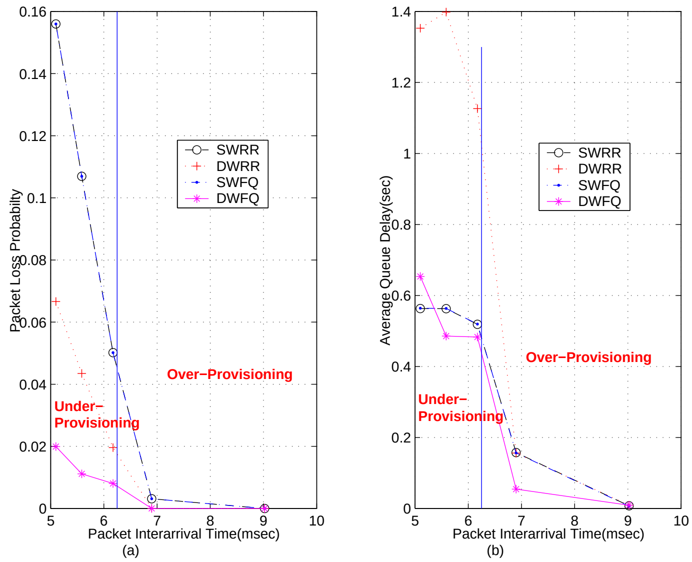

Fig. 7. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for loss-sensitive class.

illustrate for the over-provisioned scenario, while they exhibit an up to 50% degradation in performance for the under-provisioned case. This result is expected, given the fact that this class receives the lowest priority under the proposed scheme.

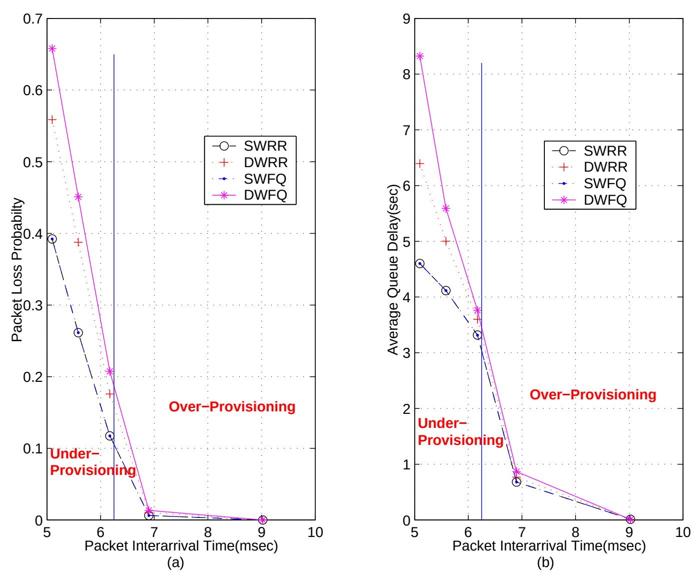

Fig. 8. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for best effort class.

The main conclusions are: (i) that the dynamic scheduling algorithms perform as well as their static counterparts for all classes in the over-provisioned scenario and (ii) they significantly *outperform* the static policies for the delay and loss sensitive classes under the remaining scenarios.

We examine next the effect of the pre-specified threshold parameters (θ_1, θ_2) on the performance of the proposed scheme. It is easy to see that the higher the threshold θ_1 (for the delay-sensitive class), the better performance our proposed scheduling algorithms would achieve for that particular class. We thus briefly turn our attention to the loss-sensitive class, and explore the effect of θ_2 on the various classes. Two values are investigated, namely, $\theta_2 = 0.90$ and 0.95. Figures 9 (a) and 9 (b) show that the proposed scheduling algorithms achieve a better performance for the underlying loss-sensitive class, as expected, but more importantly the performance of the remaining classes is not significantly affected. A more extensive evaluation of the effect of the thresholds is currently under study.

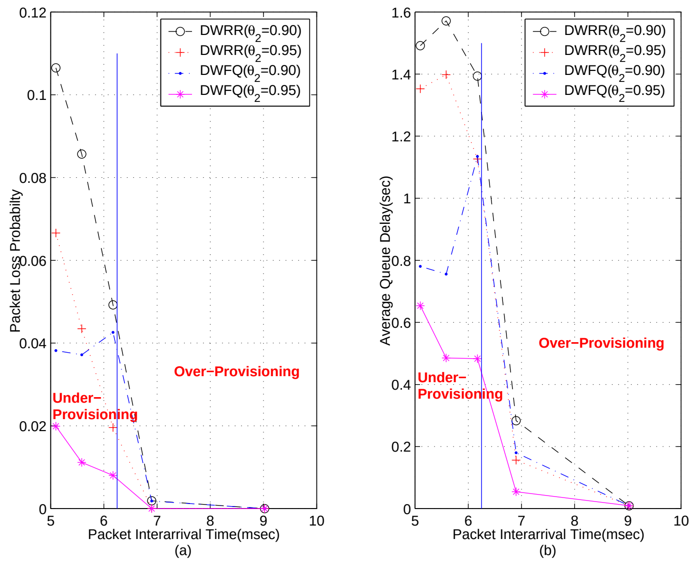

Fig. 9. Comparison of DWFQ and DWRR on (a) packet loss probability and (b) average queue delay with $\theta_2 = 0.90$ and $\theta_2 = 0.95$.

We evaluate next how the interval over which traffic patterns change over time affects the performance of the proposed scheme in the under-provisioned case. We do this by defining an interval, called the pattern change interval (PCI), over which the traffic load distribution for each class changes.

In this evaluation all the other tuning parameters remain fixed for the duration of the experiment. The PCI ranges for 10 to 90 sec, while the adaptive window size is 1 sec. Therefore, the dynamic policies have enough time to adapt to the changing traffic conditions and thus deliver good performance.

It can be seen from Figures 10 (a) and 10 (b), that the performance of DWFQ is not affected at all by changes in the PCI for the delay-sensitive class and similar conclusions can be reached to a large extent for the loss-sensitive one. On the other hand, the DWRR class exhibits an improving performance over larger PCIs for the delay-sensitive class and a degrading one over the loss sensitive (see Figure 11). Finally, both policies exhibit some variability for the best effort class over the range of PCI (see Figure 12). In terms of relative performance with respect to the static policies, the conclusion previously reached still apply. A topic that is currently under investigation is at what point the performance of the proposed dynamic allocation scheme degrades significantly as a function of PCI, or in other words for what values of the ratio of W/PCI the performance exhibited is deemed satisfactory.

Next, we briefly report on our investigations regarding the composition of the traffic and its effect on performance.

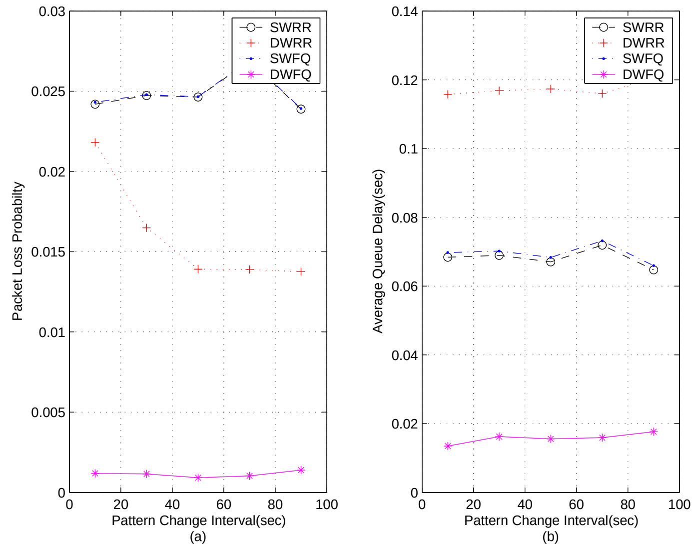

Fig. 10. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for delay-sensitive class over pattern change interval.

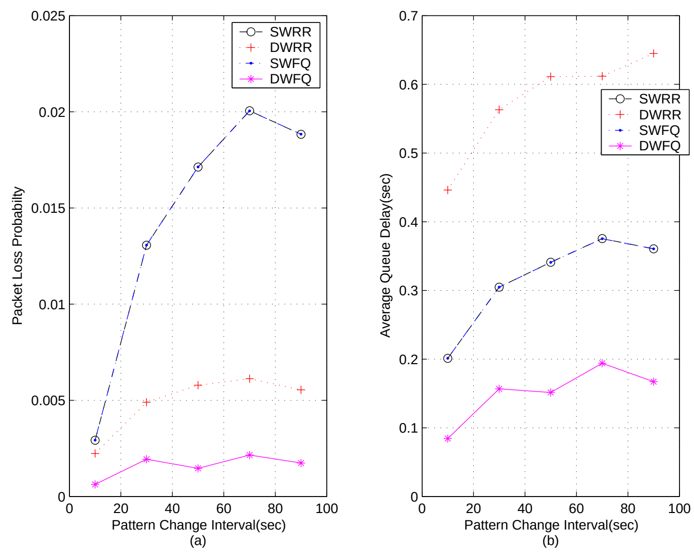

Fig. 11. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for loss-sensitive class over pattern change interval.

The benchmark is a uniform distribution over the classes (i.e., equal proportions of $1/3$, $1/3$, and $1/3$) and we also report the results for other unbalanced distribution such as those given in Table I, where over non-overlapping time intervals of length 100 sec, the composition of traffic follows those proportions; namely, at some point in time approximately 17% of the total traffic would come from the delay-sensitive class, an equal proportion from the loss-sensitive class and the remaining from the best effort class, while at another point in time, approximately 67% of the traffic would belong to the delay-sensitive class, and 17% to the other two classes. The various traffic load distributions examined are given in Table I, where $\sigma =$(max proportion/min proportion). A plot of the distribution of traffic as it is varying over time is given in Figure 13, that provides some insight into the changing nature of

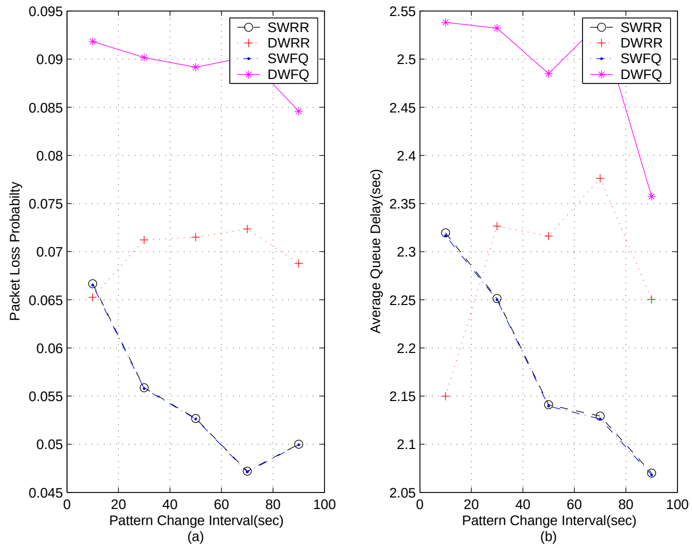

Fig. 12. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for best effort class over pattern change interval.

traffic.

TABLE I

INITIAL PROPORTIONS OF TRAFFIC DISTRIBUTION AND CORRESPONDING σ VALUES.

	Delay-Sensitive Class	Loss-Sensitive Class	Best Effort Class
$\sigma = 1$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
$\sigma = 2$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$
$\sigma = 3$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{3}{5}$
$\sigma = 4$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{2}{3}$

Figure 15 (a), shows that the performance of the dynamic algorithms is superior to that of the static ones for unbalanced traffic distributions for the delay-sensitive class, regarding losses. It is also clear that the DWFQ enjoys a large margin over its competitors with respect to both performance metrics. On the other hand, the performance of DWRR with respect to delays is rather problematic, due to the normalization step used. Analogous conclusions can be reached for the loss-sensitive case for unbalanced distributions as Figure 16 shows. For the best effort class, due to its low priority given by the dynamic scheme, the static algorithms outperform their dynamic counterparts, although the differences become smaller for more unbalanced load distributions.

In Figures 17 (a) and (b) the performance of the DWRR and the DWFQ policies is inferior to their static counterparts, as expected, due to the allocation of a large percentage of the available bandwidth to higher priority classes.

Overall, for very unbalanced composition of the traffic, our proposed scheme significantly outperforms the static algorithms, due to its flexibility and adaptiveness over time.

Finally, we briefly report on preliminary results obtained about the effect on performance of the adaptive window

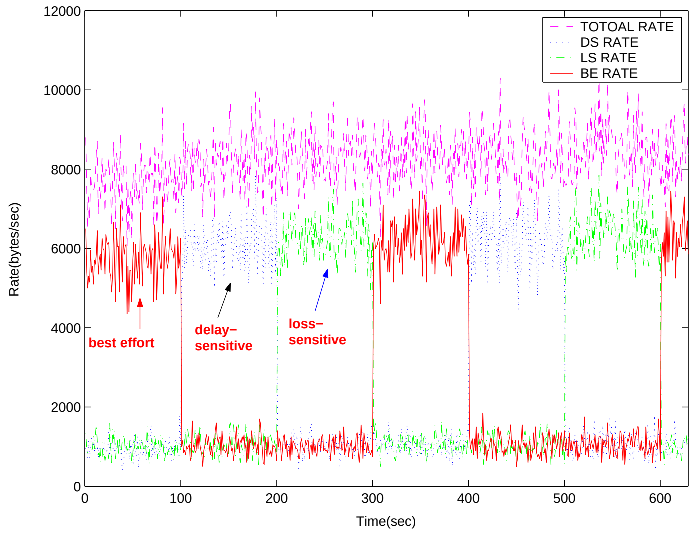

Fig. 13. Total traffic rate and traffic rates for different classes when $\sigma = 4$.

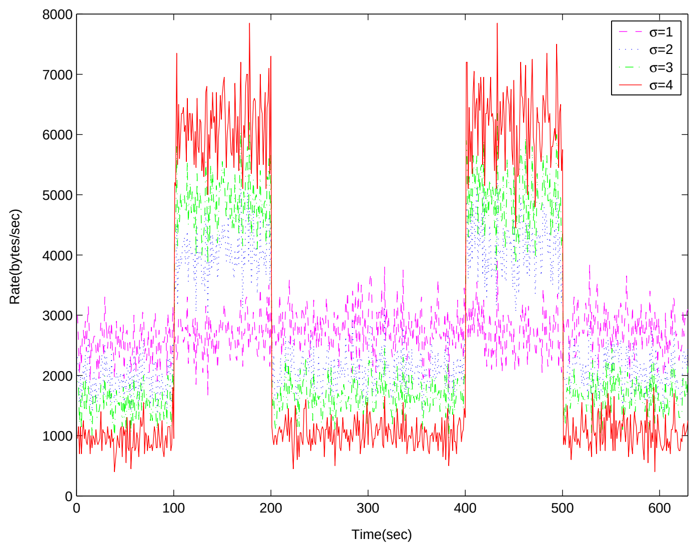

Fig. 14. Traffic rates for delay-sensitive class with different σ.

W. In [11], the adaptive window was chosen as a multiple of the measurement slot τ. Thus, the choice of τ is a key factor in determining the accuracy of the obtained measurements, which in turn affect the performance of the proposed scheme. If the value of τ is too small, it can capture the burstiness of the coming traffic and result in over-allocation of resources. The QoS requirements would be satisfied, but the computational requirements of our scheme become quite high. On the other hand, if the value of τ is too large, the measurements cannot capture well the fluctuations in the traffic processes, thus compromising performance.

It should be noted that in a rapidly changing traffic environment a near-optimal selection of τ becomes crucial; however, this represents a topic of current research. In this paper, the objective is to demonstrate the effect of τ on performance.

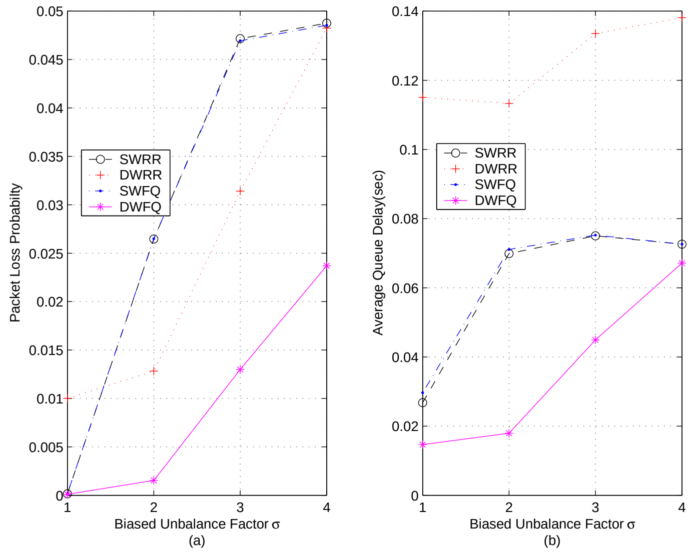

Fig. 15. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for delay-sensitive class as σ changes.

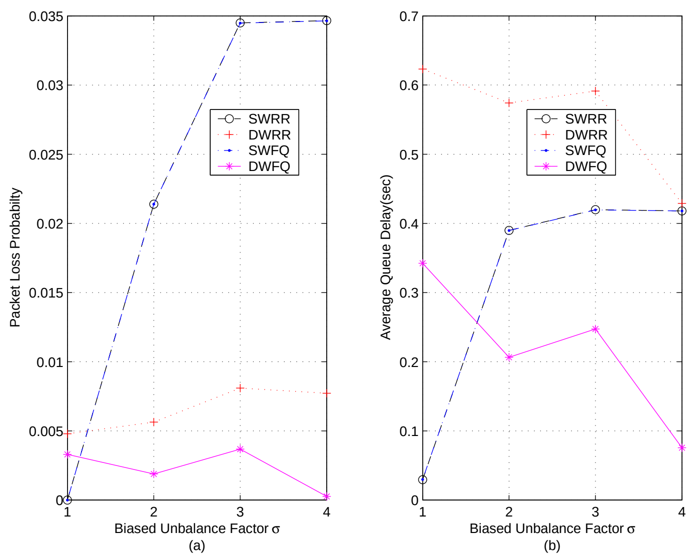

Fig. 16. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for loss-sensitive class as σ changes.

In Figures 18 (a) and 18 (b), the loss probability and average queue delay increase as τ increases, which is consistent with our intuition and our analysis. Moreover, the confidence intervals for the performance of the proposed scheme, and in particular for the DWFQ policy, are significantly narrower than those for the static algorithms. This reflects the more robust nature of our scheme.

Figures 19 (a) and 19 (b) show analogous results for the loss-sensitive class. However, for windows larger than 1.5 sec the delay performance of the DWFQ policy becomes inferior to that of the static algorithms. This indicates that the choice of W (and its relationship with the time scale over which traffic patterns change) is important for delivering the required QoS.

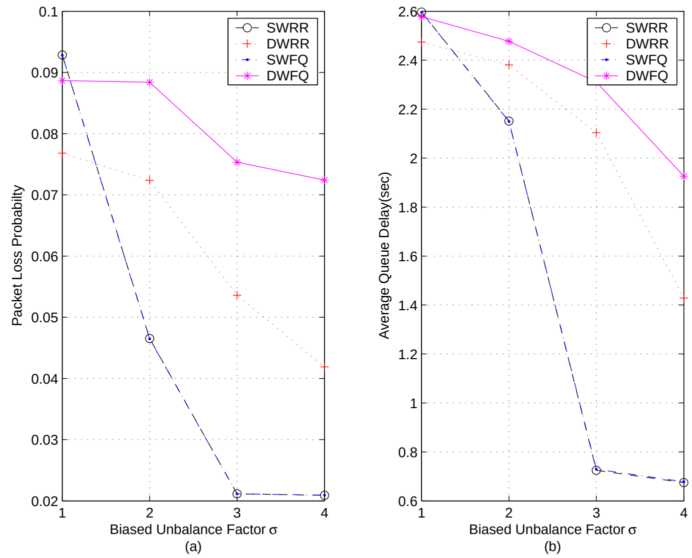

Fig. 17. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for best effort class as σ changes.

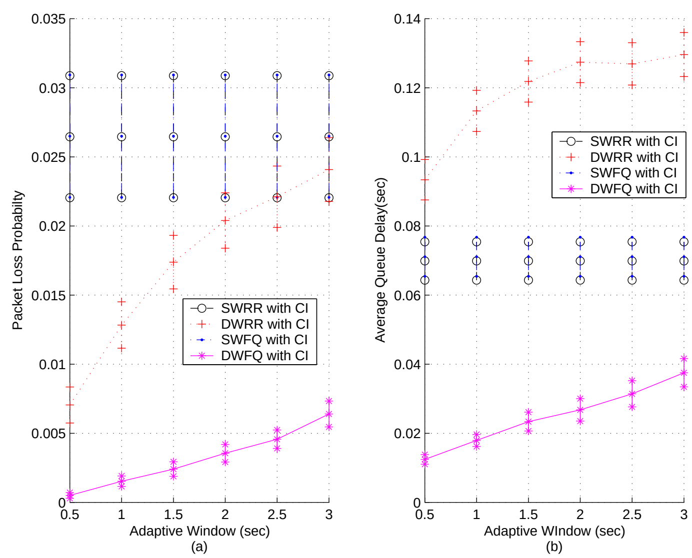

Fig. 18. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for delay-sensitive class as W changes.

Finally, in Figures 20 (a) and 20 (b) it can be seen that for small values of τ, the over-allocation for the higher priority classes results in an under-allocation for the best effort class. As τ increases, the proportion of bandwidth reserved for the higher priority classes gradually shifts to the best effort class, which yields a better QoS performance.

IV. Conclusions

In this paper a new scheduling scheme, that combines measurement estimation with standard scheduling algorithms, is introduced that provides QoS guarantees to different traffic classes under the DiffServ mechanism. The proposed scheme, due to its *dynamic and adaptive* nature exhibits a superior performance to static algorithms. Two

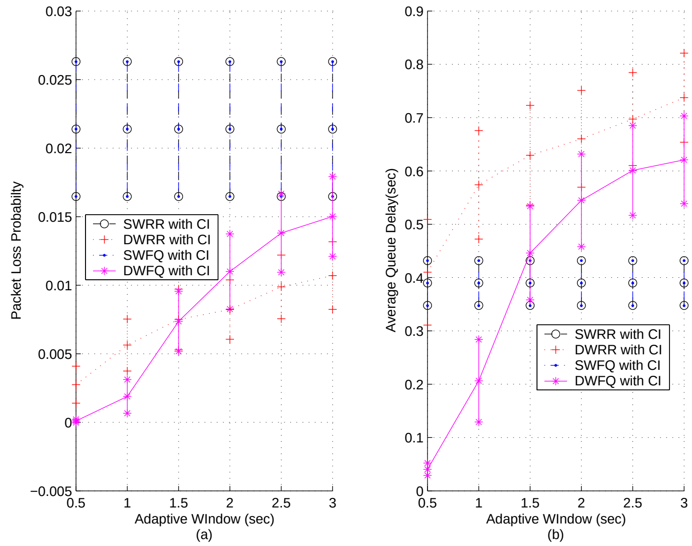

Fig. 19. Comparison of scheduling algorithms on (a) packet loss probability and (b) average queue delay for loss-sensitive class as W changes.

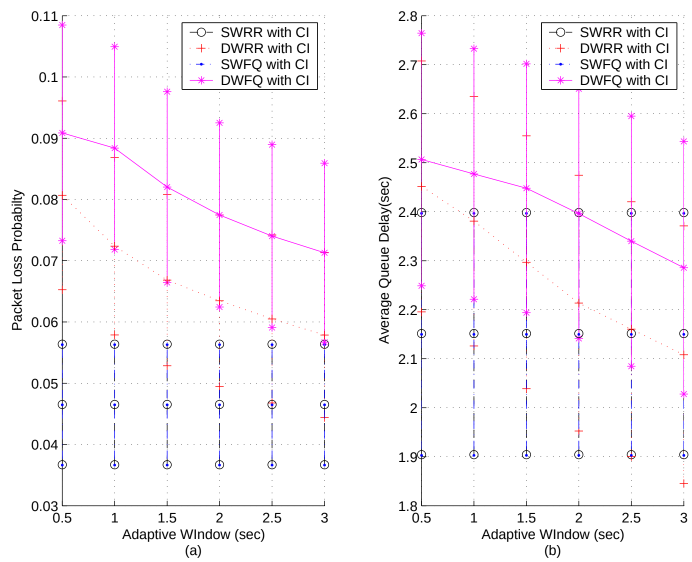

Fig. 20. Comparison of scheduling algorithms on (a) packet loss probability and b) average queue delay for best effort class as W changes.

dynamic algorithms DWFQ and DWRR have been investigated and the former outperforms the latter in almost all scenarios examined, due to its inherent flexibility on determining the weights. Our scheme proves robust to the underlying load distribution of the traffic classes and to the changing nature of traffic characteristics over time.

The proposed scheme relies on two, externally set parameters (e.g. by the ISPs) (θ_1, θ_2), that determine the relative importance of the traffic classes. It also utilizes an adaptive sampling window, whose optimal determination is a topic of current work.

ACKNOWLEDGMENTS

The work of this paper was done while the last two authors were participants at SAMSI's Network Modeling for the Internet program in the Fall of 2003.

REFERENCES

[1] D. Xuan. Part i: Introduction. NA6.ppt. [Online]. Available: http://www.cis.ohio-state.edu/ xuan/courses/679/

[2] W. Wei. Qos and video delivery. QoS_Video.ppt. [Online]. Available: http://www-inst.eecs.berkeley.edu/ ee290t/

[3] A. Kulkarni and Y.-A. Chen. (2000) QoS (Intserv and Diffserv). presentation.ppt. [Online]. Available: http://www.cs.buffalo.edu/ qiao/cse620/present_2000/

[4] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, *Overview and Principles of Internet Traffic Engineering*, IETF, RFC 3272, May 2002.

[5] F. L. Faucheur and W. Lai, *Requirements for Support of Diff-Serv-aware MPLS Traffic Engineering*, IETF draft, June 2002.

[6] Frank Kelly, "Notes on Effective Bandwidth," in *Stochastic Networks: Theory and Applications*, F. P. Kelly, S. Zachary, and I. Ziedins, Eds. Oxford University Press, 1996, pp. 141–168.

[7] X. Xiao and L. Ni, "Internet qos: the big picture," *IEEE Network Magazine*, vol. 13, pp. 8–18, Mar 1999.

[8] L. Breslau and S. Jamin, and S. Shenker, "Comments on the Performance of Measurement-based Admission Control Algorithms," in *Proc. of IEEE INFOCOM*, 2000, pp. 1233–1242.

[9] S. Jamin and S. J. Shenker, and P. B. Danzig., "Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service." in *Proc. of IEEE INFOCOM*, 1997.

[10] E. Knightly and N. Shroff, "Admission Control for Statistical QoS: Theory and Practice," *IEEE Network*, vol. 13, no. 2, pp. 20–29, 1999.

[11] J. Qiu and E. Knightly, "Measurement-based Admission Control with Aggregate Traffic Envelopes," *IEEE/ACM Trans. Networking*, vol. 3, pp. 199–210, April 2001.

[12] Z. Zhang, C. Sanchez, B.Salkewicz, and E.Crawley, "Quality of Service Extensions to OSPF," *Work in Progress*, Internet Draft, Sept. 1997.

[13] R. Buerin, S.Kamat, A. Orda, T. Przygienda, and D. Williams, "QoS Routing Mechanisms and OSPF Extensions," *Work in Progress*, Internet Draft, March 1997.

[14] J. Yan, "Adaptive configuration of elastic high speed multiclass networks," *IEEE Communication Magazine*, pp. 116–120, May 1998.

[15] Q. Hao, S. Tartarelli, and M. Devetsikiotis, "Self-Sizing and Optimization of High Speed Multiservice Networks," in *Proceedings of IEEE GLOBECOM*, vol. 3, 2000, pp. 1818–1823.

[16] S. Nelakuditi, S. Varadarajan, and Z. Zhang, "On localized control in QoS routing," *IEEE Transactions on Automatic Control*, vol. 47, no. 6, pp. 1026–1032, June 2002.

[17] S. Nalatwad and M. Devetsikiotis, "Self-Sizing Networks: Local vs. Global Control," To be presented at IEEE ICC 2004, Paris, France, June 2004.

[18] H. Wang, C. Shen, and K. G. Shin, "Adaptive-Weighted Packet Scheduling for Premium Service," in *Proceedings of the IEEE ICC*, 2001.

[19] A. Chandra, W. Gong, and P. Shenoy, "Dynamic Resource Allocation for Shared Data Centers Using Online Measurements," in *Proceedings of ACM/IEEE Intl Workshop on Quality of Service (IWQoS)*, 2003, pp. 381–400.

[20] S. K. A. Demars and S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm," in *ACM SIGCOMM*, 1989, pp. 3–12.

[21] A. W. Moore, "Measurement-based management of network resources," University of Cambridge, England, Tech. Rep. UCAM-CL-TR-528, April 2002.

[22] B. Ryu and S. Lowen, "Fractal Traffic Models for Internet Simulation," in *IEEE International Symposium on Computers Communications*, France, 2000, pp. 271–370.

Appendix: Dynamic Resource Allocation with Optimization

We provide a brief outline of our analytical framework based on an implicit pricing scheme.

Under DiffServ, since the higher class of traffic is guaranteed a better QoS than the lower classes, it should be charged a higher price per unit of resource utilization accordingly. In this way, the profit is maximized as the maximal utilization of resource has been reached. Meanwhile, minimizing the queue delay for all classes of traffic should be also taken into account. If we associate the queue delay with "cost"(or actual monetary cost), the failure of better QoS provision for the higher class would be penalized with a higher cost per unit of time than the lower classes. From the perspective of the network provider, this delay-involved cost should be minimized. If we combine the above consideration together, we could obtain a unified maximization formulation of network provider's profit as follows:

$$\max \sum_{i=1}^{3} (p_i \phi_i C - b_i D_i) \tag{6}$$

where

p_i – the price of per unit of resource utilization of ith class and $p_i > p_j$ if $i > j$

ϕ_i – the share of the bandwidth for ith class

C – the capacity of the router

b_i – the delay cost of per unit of time for ith class and $b_i > b_j$ if $i > j$

D_i – the average queueing delay of ith class

The maximization of the objective function is subject to the following constraints:

$$\sum_{i=1}^{3} \phi_i \leq 1$$

$$\phi_i \geq EB_i / C$$

$$D_i \leq d_i$$

where d_i is the desired queueing delay of ith class and EB_i is effective bandwidth of ith class.

EB_i could be obtained by measurement estimation, while D_i could be derived from the same fluid model adopted in [19]. The specifics of D_i are not covered in this appendix, but it could be concluded that the relationship between D_i and ϕ_i is nonlinear.

Therefore, the maximizing the objective function becomes a nonlinear programming optimization problem with inequality constraints. This problem could be solved by Khun-Tucker conditions, meanwhile the concavity and differentiability of the objective function should be also considered and discussed. The derivation of the optimal solution of this objective function and simulation results will be included in our next upcoming paper.