Proportional QoS Provision: A Uniform and Practical Solution

Yang Chen[1], Mounir Hamdi[2], Danny H.K. Tsang[3] and C. Qiao [1]
[1]CSE Department, SUNY Buffalo
[2]Computer Science Department, Hong Kong University of Science and Technology
[3]Electrical and Electronic Engineering Department, Hong Kong University of Science and Technology
Email: hamdi@cs.ust.hk

Abstract— **The *proportional Service model* is receiving a lot of attention as an attractive model for providing differentiated services on the Internet. In particular, this model is controllable, able to provide the "tuning knobs" for network operators to quantitatively differentiate the quality-of-service (QoS) of different classes, and lends itself naturally to simple pricing schemes. In this paper, we focus on the issue of how to practically implement such a QoS differentiation scheme at high-speed routers using efficient buffer management and packet scheduling mechanisms. We first propose a *uniform* scheduler. Unlike previously proposed schedulers which can be used only for a single QoS metric, our scheduler is suitable for various QoS metrics. We then introduce a new packet dropping mechanism with an active counter resetting scheme that compare favorably with previous schemes. Finally, we develop an original and simple approach for the integration of absolute QoS constraints with the proportional differentiation paradigm.**

I. INTRODUCTION

Recently, a refinement of relative differentiated QoS, called proportional differentiation model, is receiving a lot of attention from the academic and industrial communities [1].It provides the network operators with adjustable and quantitative QoS differentiation between service classes, which cannot be achieved with other relative differentiation models, such as strict prioritization, capacity differentiation, or price differentiation. With the proportional differentiation model, the network operators are able to quantitatively adjust the differentiation levels between classes based on the pre-specified factors. Different QoS metrics [1–4] in various network infrastructures [5, 6] have been investigated using this scheme . Unfortunately, those efforts are independent from each other. This hinders their practical implementation on high-speed switches/routers as each QoS metric needs its own hardware and control units. In addition, because of the simplicity of the relative QoS model, absolute QoS guarantees cannot be realized using a proportional differentiated model directly.

In this paper, we expand and improve upon previous research efforts with the aim to provide a simple and effective architecture for proportional QoS provision with/without absolute QoS constraints. One of our goals is to simplify the practical implementation of proportional QoS provision on high-speed routers. The contributions of this paper can be summarized as follows:

- A uniform scheduler which is suitable for various QoS metrics is proposed.
- A new packet dropping mechanisms, namely, PLR with an active resetting which overcome the complexity of previous droppers' implementations and operation, is proposed and evaluated.
- Based on the idea of constraints relaxation proposed in [8], we develop a simple scheme to achieve absolute QoS guarantee of two QoS metrics of interest: packet delay and loss rate.

The rest of the paper is organized as follows. In Section 2, we briefly present the proportional differentiation model and previous work on various service differentiations. In Section 3, our practical packet scheduling and dropping framework is proposed. How to integrate the absolute QoS constraint on delay and loss in our framework is explained in Section 4. Extensive simulation results are given to show the effectiveness of our practical framework. Section 5 concludes this paper.

II. PROPORTIONAL QOS

A. Proportional QoS Model

In a relative QoS model, we can only guarantee that the traffic from a higher priority class will receive no worse service than the traffic from a lower priority class [7]. As an improvement, in the proportional differentiation model, the service differentiation can be quantitatively adjusted to be proportional to the differentiation factors that a network service provider sets beforehand. If q_i is the QoS metric of interest and s_i is the differentiation factors for class i, in the proportional differentiation model, we should have:

$$\frac{q_i}{q_j} = \frac{s_i}{s_j} \qquad (i, j = 1 \ldots N) \tag{1}$$

For example, in a packet network, assume that l_1, l_2 are the packet loss rates for class 1 and 2 respectively. If s_1 is 1 while s_2 is 2, then we should have $l_1/l_2 = 1/2$, which means the packet loss rate of class 2 should be twice that of class 1.

Because the long term average is not quite meaningful when the traffic is bursty, it is desirable that the proportional differentiation model holds over not only long time periods but also short time periods as well. That is, the proportional differentiation equation (1) should hold within a short time period τ, which is called *monitoring timescale* in [1]:

$$\frac{\bar{q}_i(t, t+\tau)}{\bar{q}_j(t, t+\tau)} = \frac{s_i}{s_j} \tag{2}$$

where $\bar{q}_i(t, t+\tau)$ is the average QoS metric in the time period τ. This service model is general enough in that the quality differentiation between traffic classes can be defined as a function of various QoS metrics.

This research work was supported in part by grant from the Hong Kong Research Grants Council under the grant HKUST6202/99E.

0-7803-7400-2/02/$17.00 © 2002 IEEE

B. *Previous Results*

The first QoS metric discussed in a proportional QoS provision is average packet delay [1]. A Waiting Time Priority (WTP) scheduler has been proposed in order to achieve proportional average delay differentiation. The work in [10] improves WTP scheduler's performance when traffic load is not heavy. There is no particular requirements on packet droppers in a proportional average packet delay provision - any simple dropper dropping packets from the queue's tail is applicable.

A more recent work [2] extended the proportional QoS model to another important QoS metric: packet loss rate. Two droppers: PLR(M) and PLR(∞) are proposed in order to provide proportional packet loss rate. Note that there is no special requirements on packet schedulers here, a simple First Come First Serve (FCFS) scheduler is applicable.

The performance of multimedia applications, e.g., IP telephony, does not depend on average packet delay but rather on the probability that the packet delay exceeds a certain threshold [9]. This probability called deadline violation probability. The proportional QoS model has been applied to it in [3].

Unfortunately, these previous research efforts are independent from each other. For each QoS metric, a different router mechanism is needed to achieve proportional differentiation. This means we need different hardware and control units for different QoS metrics. In order to make the proportional QoS provision practical, we should decrease the complexity of their implementation on a router as much as possible. This is the key motivation of our proposed uniform scheduler.

III. PRACTICAL FRAMEWORK

A. *Time-Based Uniform Schedulers*

Scheduler	**Mathematical representation**
WTP	$Max\{(t_{current} - t_{i.arrival})/\delta_i\}$
EDD	$Min\{t_{i.arrival} + d_i\}$
FCFS	$Max\{t_{current} - t_{i.arrival}\}$
$t_{current}$: Local time when a packet need to be scheduled $t_{i.arrival}$: class i packet's arrival time d_i: Deadline for class i	

TABLE I

MATHEMATICAL FORM OF DIFFERENT SCHEDULERS.

The mathematical representation for each scheduler is summarized in Table I. We can see that it is easy to combine the WTP scheduler with an ordinary FCFS because by setting all δ_i to be 1. Noticing the equivalence between the different "decisions" below, we can unify the three schedulers.

1. $Min\{t_{arrival} + d\} = Max\{-t_{arrival} - d\}$
2. $Max\{-t_{arrival} - d\} = Max\{t_{current} - t_{arrival} - d\}$
3. $Max\{t_{current} - t_{arrival} - d\}$

$\iff Max\{\frac{t_{current} - t_{arrival} - d}{\delta}\}(\delta = 1)$

As a result, we produce a uniform scheduler of the following form: $Max\{\frac{t_{current} - t_{i.arrival} - d_i}{\delta_i}\}$ which can operate exactly as the three schedulers discussed above. The packet of class i which has this maximum value will be scheduled. There are two adjustable parameters in this scheduler: δ_i and d_i. The parameter setting and corresponding scheduler is as follows:

1. d_i: 0, δ_i: proportional factor for class i , WTP scheduler
2. δ_i: 1, d_i: deadline for class i, EDD scheduler
3. δ_i: 1, d_i: 0, FCFS scheduler

The WEDD scheduler is actually not a time-based but measurement-based scheduler, hence, it cannot be realized in this uniform form.

B. *Droppers/Schedulers with Resetting*

All the previous measurement-based droppers/schedulers require a set of counters. There are two problems associated with using counters. The first is counter overflow. Various approaches are used in the previous research to solve this problem. In [2], the counters are simply reset when the overflow occurs. In [3], the counters value will be multiplied by a value α, $(0 < \alpha < 1)$ whenever the counters are updated. The second problem is that if we make the packet dropping/scheduling decision based on a long-term measurement, we may achieve the proportional differentiation over a long time period but violate the proportional differentiation in a short time period. In order to adapt to the load fluctuation, we should make the drop decision based on a limited recent history. This is the reason that PLR(M) is more adaptive to the load fluctuation than PLR(∞). However, using PLR(M), a cyclic queue will be maintained for the loss and arrival information of each class in the recent M packet arrivals. As a result, an extra interior tag is required for cyclic queue updating after each arrival/drop. This will increase the complexity of its hardware implementation.

Here, we will apply an active resetting process to the counters. As a result, we will have a simpler proportional loss rate provision scheme which is adaptive to load fluctuations. The detailed algorithm is given below:

```
1. Set two counters $A_i$ and $D_i$ to record packet arrivals and packet droppings of class $i$;
2. When a packet of class $i$ arrives,
if (there is no free space in the buffer)
    a packet from class $j$ having minimum $D_j/A_j\sigma_j$
    among the $N$ classes will be dropped, the newly
    arrived packet enters the queue, $D_j + 1$, $A_i + 1$;
else
    this packet entering the queue, $A_i + 1$;
3. Update $l_i = A_i/D_i,\ (i = 1 \ldots N)$.
if ($Max\{l_i/l_1 - \sigma_i/\sigma_1 | i = 2 \ldots N\}$ is less than the error threshold $ERROR$)
    go to step 1;
else
    go to step 2.
```

By resetting the counters whenever equation (2) is satisfied within a limited deviation, we can achieve proportional differentiation over a long time period as well as in a short time period because it makes its dropping decision based on a recent history.

In particular, the proportional dropper is more adaptive to load fluctuation.

IV. ABSOLUTE QOS CONSTRAINTS

There are many types of traffic which require strict QoS guarantees. For example, a real-time application puts a stringent requirement on packet delay. A business data transaction cannot bear a loss rate exceeding a given threshold. Since the proportional differentiation model is a relative QoS model, absolute QoS constraints cannot be provided by itself [7]. If network service providers or operators want to provide service to those applications with absolute constraints in a proportional paradigm, some modifications are required.

Joint Buffer Management and Scheduling (JoBS) [8] has been proposed as a solution to provide absolute QoS constraints on two QoS metrics of interest: average packet delay and packet loss rate. At the same time, JoBS tries to maintain the proportional differentiation among service classes. In order to provide absolute service bounds, it applied a method called constraint relaxation. Absolute QoS constraints have higher priority than proportional QoS constraints. When there are conflicts between constraints, the constraints with lower priorities will be relaxed.

A. Packet Delay

Most of the computation complexity in JoBS is caused by the need to predict average packet delay. Here we use a delay threshold d to replace absolute average delay to reduce the complexity in operation. We believe this is also more practical because a deadline in delay is much meaningful than an average delay for a time-stringent application. For example, a packet of IP telephony will be useless if it arrives at the destination violating its deadline although the average packet delay might be kept within a given limitation.

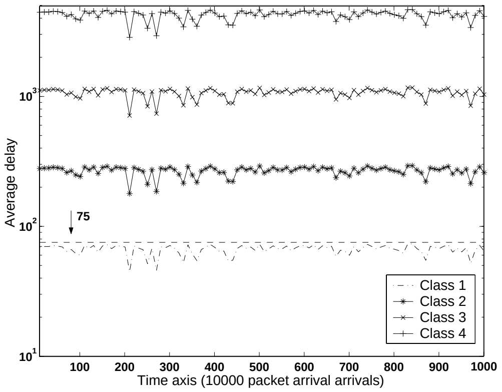
Providing absolute delay guarantee in the proportional delay provisioning model should operate in the following manner: when a packet needs to be forwarded, all the packets violating their deadline are dropped first, after that, a WTP scheduler will finish the scheduling work. Dropping deadline-violating packets guarantees the delay constraints while the WTP scheduler provides proportional delay to those classes which do not violate their delay constraints.

B. Loss Rate

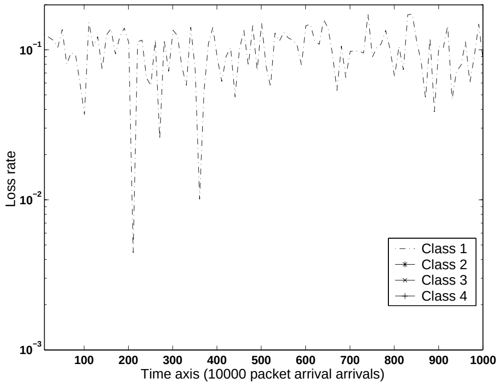
In JoBS, when the buffer experiences overflow, an optimization criteria with the objective of “not violating the absolute loss rate constraint” will decide the change in the loss rate for each class. Here, we propose an effective change to the original PLR droppers. When a buffer overflows and a run-time measurement is done, if a particular class violates its loss rate bound, its weighted lose rate $D_i/A_i\sigma_i$ is set to be 1. This guarantees that a packet from the classes which do not violate the loss rate bounds will be dropped. An ordinary PLR dropper then makes the final dropping decision and maintains the proportional loss rate among those classes which do not violate the loss rate bounds.

C. Packet Loss and Delay

In the previous sections, we have only illustrated the absolute QoS provisioning on packet loss or delay separately. However, in a practical implementation, these two metrics are always correlated. In fact, they are more related in our scheme in that the packets violating their deadline will be dropped. The dropped packets should be included in the total packet loss. Simply including packets violating their deadline in the total loss and combining the above two schemes will lead to an unexpected high loss rate for a service class which has an absolute delay constraint. We will illustrate this problem by a numerical example, where a simulation scenario similar to that used to evaluate JoBS’s performance is assumed. Specifically, The total buffer size is 2500; the load distribution is $\lambda_1 : \lambda_2 : \lambda_3 : \lambda_4 = 0.25 : 0.25 : 0.25 : 0.25$; we choose the proportional factors for packet loss rate as: $1 : 2 : 4 : 8$; for the delay, the factors are $1 : 4 : 16 : 64$. (These will be the default simulation scenario for the following simulations unless otherwise specified.) A deadline of 75 is set for class 1; traffic load is 1; and the packet loss rate and delay is measured every 100,000 packet arrivals.



(a) Average packet delay.



(b) packet loss rate.

Fig. 1. Problem in absolute delay constraint provision.

From Figure 1(a), we can find that the proportional and absolute packet delay can be provided over even a short time period, which is preferred. However, as to packet loss rates shown in Figure 1(b), since a large number of class 1 packets get dropped

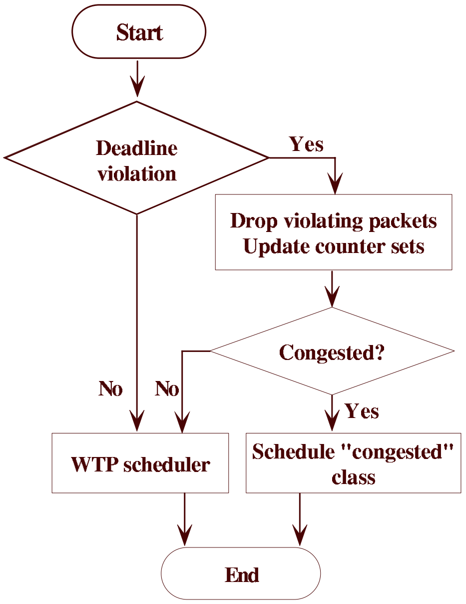


Fig. 2. Modified packet scheduling

due to deadline violation when the traffic is relatively high, this keeps the buffer from overflowing, which means no packet from other classes will be dropped. This problem does not occur in JoBS because when it predicts a high average delay, it will not only adjust the loss rate for each class, but the service rate is also changed in order to provide a higher service rate for a class with absolute delay constraints. However, in the scheme described in the previous section, a WTP scheduler is used, which does not intentionally increase the service rate of class 1.

Since the only solution is to increase the service rate for a class with absolute delay constraint when some of its packets are dropped due to deadline violation, we will modify the previous scheduling schemes by following the line of thought in [3]. More specifically, if we set d_i for a particular class i, a safety margin Δ_i (usually, $\Delta_i = d_i/10$) is also set for this class. When a packet of class i is dropped due to deadline violation, if this class is still backlogged with the first packet having a deadline $t_a + d_i < t + \Delta_i$, this class is said to be in “congested” mode. A packet from this class is scheduled directly instead of using a WTP scheduler. Hence, we can increase the service rate of class i so that not too many packets will be dropped. The modified scheduling algorithm is illustrated in Figure 2 and detailed simulation results are given in the next section showing the effectiveness of this joint delay and loss rate managing scheme.

D. *Simulation results*

In this set of simulations, we will show the integrated proportional and absolute QoS provision comprehensively. Absolute delay and loss constraints are monitored jointly. In this set of simulations, the traffic load changes from 1 to 1.1 after 30,000,000 packet arrivals and changes again to 0.9 after 60,000,000 packet arrivals. All the other simulation scenarios are the same as in the previous section.

At first, we put an Absolute Loss (rate) Constraint (ALC) on class 1 to be 0.001. The simulation results is shown in Figure 3(a) to Figure 3(c). Then we also test our scheme’s performance when there is an Absolute packet Delay Constraint on class 1 set to 75.

Note that Besides being able to provide absolute QoS constraints, JoBS has an extra advantage over the PLR(∞)/WTP combination in that JoBS is more adaptive to load fluctuation. In Figure 3, we can see that our scheme is also adaptive to the load fluctuation. The absolute QoS constraints are strictly maintained with some relaxation of the proportional factors. There is little oscillation in the proportional relationship at the traffic load transition points.

V. CONCLUSION

One of the contributions of this paper is the proposed practical and uniform framework for packet scheduling/dropping implementation in a proportional differentiation service model. A uniform scheduler which is versatile for different QoS metrics has been proposed. A new dropper, PLR with active resetting has also been proposed to compare with the previous droppers PLR(M) and PLR(∞).

We have also used constraint relaxation to provide absolute QoS constraints in a proportional paradigm. But unlike the previous work in JoBS [8], we have used deadline to replace the absolute average delay constraint. This not only decreases the computational complexity greatly but also makes our scheme more suitable for practical traffic requirements.

We have found that our scheme performs well in terms of achieving proportional and absolute QoS provisioning over even short time periods. In addition, our scheme is also adaptive to the traffic fluctuation.

REFERENCES

[1] C. Dovrolis, D. Stiliadis, and P. Ramanathan, “Proportional Differentiated Services: Delay differentiation and Packet Scheduling,” In *Proceedings of ACM SIGCOMM’99,* pp. 109-120, Aug. 1999.

[2] C. Dovrolis and P. Ramanathan, “Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping,” In *Proceedings of IWQoS,* pp. 52-61, 2000.

[3] S. Bodamer, “A New Scheduling Mechanism to Provide Relative Differentiation for Real-Time IP Traffic,” In *Proceedings of IEEE GLOBECOM 2000,* pp.646 -650, Vol.1.

[4] Thu Ngo Quynh, H. Karl, A. Wolisz and K. Rebensburg, “Relative Jitter Packet Scheduling for Differentiated Service,” In *9th IFIP Conference on Performance modelling and evaluation of ATM & IP Network,* 2001

[5] Y. Chen, M. Hamdi, and D.H.K. Tsang, “Providing Proportionally Differentiated Services over Optical Burst Switching Networks,” In *Proceedings of IEEE GLOBECOM 2001,* vol.3, pp.1510 -1514.

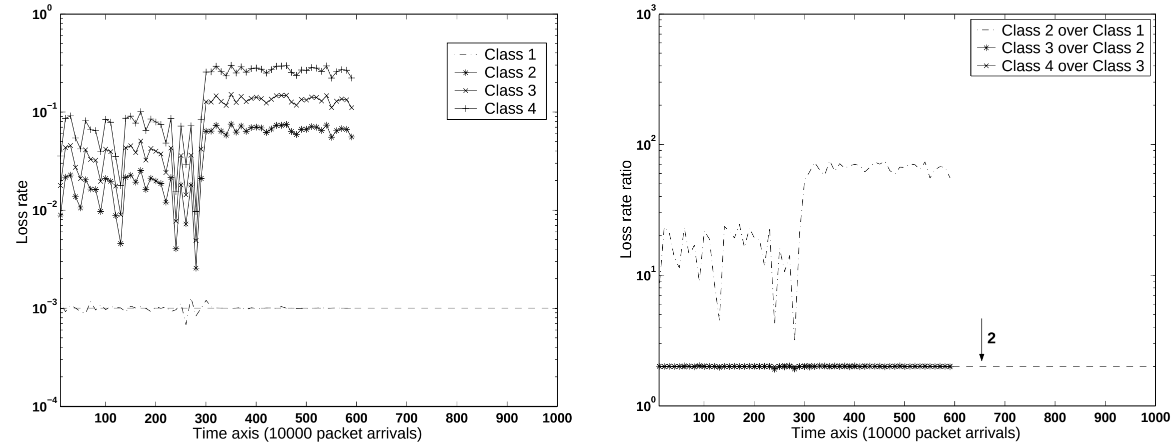
[6] Y. Chen, M. Hamdi, and D.H.K. Tsang, “Proportional QoS over WDM networks: blocking probability,” In *proceedings of IEEE Symposium on Computers and Communications,* pp.210 -216, 2001,

[7] C. Dovrolis and P. Ramanathan, “A Case for Relative Differentiated Services and the Proportional Differentiation model,” *IEEE Network,* pp.26-34, vol.13, Oct. 1999.

[8] J. Liebeherr and N. Christin, “Buffer Management and Scheduling for Enhanced Differentiated Service,” Technical Report: University of Virginia, CS-2000-24.

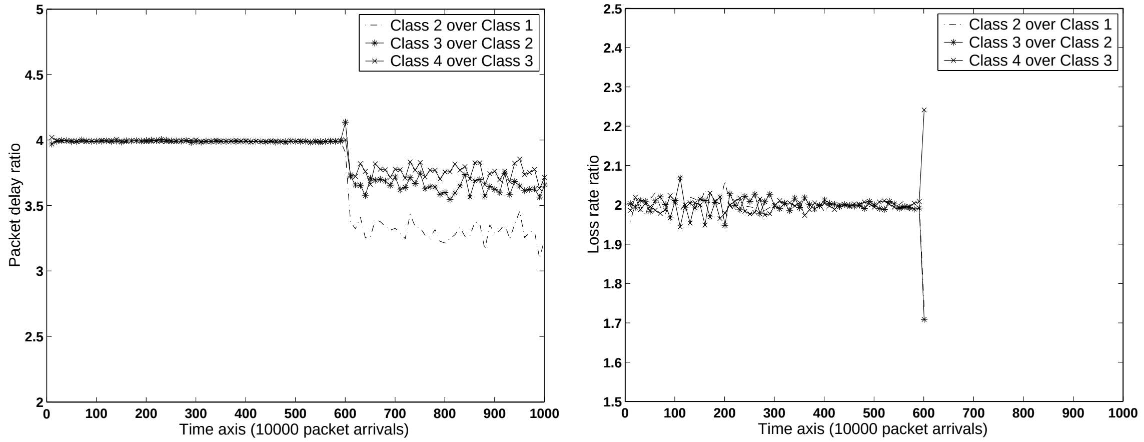
[9] C. Aras, J. Kurose, D. Reeves and H. Schulzrinne, “Real-time communication in packet-switched networks,” In *Proceedings of the IEEE,* pp.122-139, Vol.82, Jan. 1994.

[10] K. Leung, C. Lui and K. Yau, “Characterization and Performance Evaluation for Proportional Delay Differentiated Services,” In *Proceeding of IEEE International Conference on Network Protocols,* pp.295 -304, Nov. 2000.



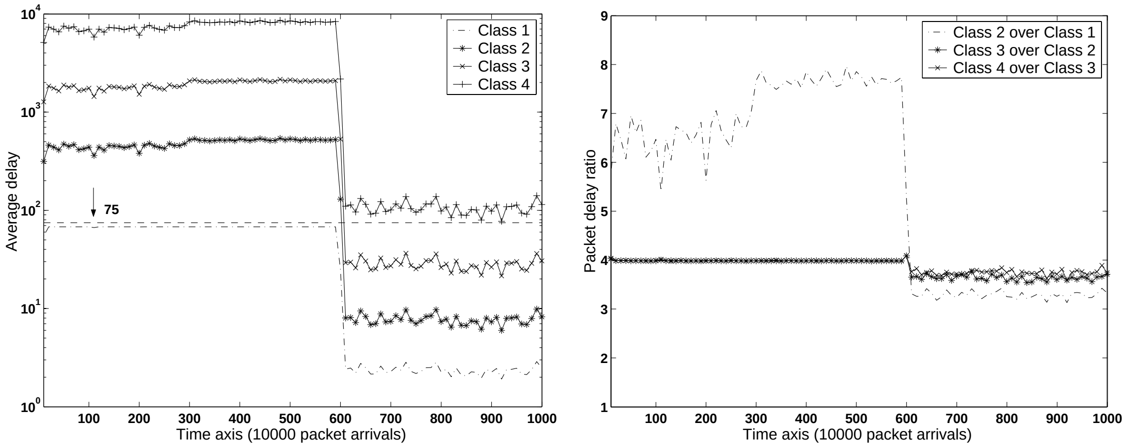
(a) Loss rate: ALC

(b) Loss rate ratio: ALC



(c) Packet delay ratio: ALC

(d) Loss rate ratio: ADC



(e) Packet delay: ADC

(f) Packet delay ratio: ADC

Fig. 3. Delay and loss rate: fluctuating traffic.