PER-CLASS QUEUE MANAGEMENT AND ADAPTIVE PACKET DROP MECHANISM FOR MULTIMEDIA NETWORKING

Mei-Ling Shyu[1], Shu-Ching Chen[2], Hongli Luo[1]

[1]Department of Electrical and Computer Engineering, University of Miami
Coral Gables, FL 33124, USA
[2]Distributed Multimedia Information System Laboratory, School of Computer Science
Florida International University, Miami, FL 33199, USA

ABSTRACT

In this paper, we propose a per-class queue management and adaptive packet drop mechanism in the routers for Internet congestion control. We model active queue management as an optimization problem and our proposed mechanism provides congestion control and fairness for different types of traffic flows. An optimal packet drop rate is obtained to maintain a relatively small queue occupancy, which provides a less queue delay delivery of packets. Simulations are conducted to compare our approach with the fixed packet drop rate approach under different traffic loads. The simulation results demonstrate that the queue occupancy changes more smoothly in our approach than in the fixed packet drop rate approach. Moreover, the queue occupancy and the packet drop rates obtained are both upper bounded, which is meaningful for providing the class-based guaranteed delay services for real-time multimedia applications.

1. INTRODUCTION

In the best-effort Internet, during the periods of congestion, TCP sources and responsive flows adjust their sending rates according to the packet drop rates in the network, while the unresponsive traffic does not reduce its sending rates. Such a highly unfair situation can lead to starvation of bandwidth and even lead to a congestion collapse [4]. As more and more multimedia applications are supported in the current Internet, quality of service (QoS) has become an important issue in the design of the routers. Hence, a new queue management approach that can protect TCP flows from UDP flows during congestion, and provide QoS for different classes of multimedia applications is desirable.

The function of queue management is to control the length or occupancy of the queue and which flows potentially occupy it [5]. Active queue management (AQM) was proposed to address the following two tasks: (1) sending a congestion signal before the router queue is full, and (2) increasing the fairness among the users when the bursty flows are subject to packet losses due to buffer overflows [2]. One of the most popular queue management mechanisms is Random Early Discard (RED) [4], where the packets are dropped before the queue is completely full, and the drop probability of the packets increases with the average queue size. Most of the RED configurations tried to obtain an optimal set of parameters based on heuristics and simulations [3]. RED can be used to control the average queuing delay in the routers and then the end-to-end delay, but the jitter of the non-bursty streams may increase. Flow Random Early Detection (FRED) [8] uses per-flow information of buffer occupancy to improve upon the fairness of RED. The Random Exponential Marking (REM) [1] aims to achieve high utilization and negligible packet losses and delays. Alternative Best Effort (ABE) [6] provided low delays for interactive applications at the expense of smaller throughputs.

In our previous work [9][10], an end-to-end framework for multimedia transmission with optimal resource utilization was proposed. In this paper, we propose a new queue management approach that can provide a small end-to-end delay for multimedia streams and at the same time protect the TCP flows from the unresponsive UDP flows for the routers. We model the general AQM as an optimization problem, and try to obtain a minimal packet drop rate that results in low queue occupancy. Compared with RED that controls the average queuing delay in the router, our per-class queue management and optimal packet drop mechanism can obtain the minimal queuing delay and hence the end-to-end delay.

The rest of the paper is as follows. In Section 2, our per-class queue management and adaptive packet drop mechanism is introduced. Section 3 gives the simulation results to illustrate the efficiency of the proposed mechanism. Conclusions are presented in Section 4.

2. OUR PROPOSED MECHANISM

2.1. Per-Class Queue Management

In per-class queuing, packets from different upstream routers that belong to the same class are enqueued into the same queue at the router. Different applications have their own QoS requirements for the presentations, such as delay, delay jitter and packet loss ratio tolerable, and thus can be classified into different classes. Per-flow queuing is another alternative for active queue management. However, it has the disadvantages such as implementation complexity and potentially lack of scalability. On the other hand, the advantage of per-class queuing is its scalability in QoS provision. It is more cost effective in terms of the maintenance of state requirements and implementation complexity. In our approach, different weights of the performance index are used for different classes to obtain different levels of queue occupancy and packet drop rates, where packets with the latest timestamp are dropped during network congestion. We measure network congestion by queue occupancy or queue length in the intermediate router. When the queue length is above a threshold value, we consider network congestion occurs. The congestion information is conveyed back to the source by packets dropping.

2.2. Adaptive Packet Drop Mechanism

Normally, a larger queue length results in a smaller packet drop ratio. However, a larger queue also means a longer queue delay, which eventually results in a longer end-to-end delay and deteriorates the playback quality for interactive multimedia applications. If the packet drop ratio is too large, the presentation quality at the client may be bad because no retransmission is provided. It is a tradeoff between the queue length and the packet drop ratio. How to strike a balance between them is important for real-time multimedia packets.

Rate control at the server alone cannot guarantee the timely delivery of the packets. Combining queue management and packet drop mechanism at the intermediate node, the transmission delay can be controlled. Hence, our objective function is to try to minimize the queue length as well as to minimize the packet drop ratio. Since the output bandwidth at a certain bottleneck router is fixed, we can assume that the service rate for the queue of a certain class is known.

Since the queue length (queue occupancy) in the next time interval is equal to the current queue occupancy plus the aggregate arriving packets and minus the output packets. That is,

$$Q_{k+1} = [Q_k + P_k - L_k]^+$$

where $[x]^+ = \max\{x,0\}$. Here, Q_k refers to the queue length at the beginning of time interval k, P_k is the packet arrival rate in the queue at k, and L_k is the output/service rate at k.

If after holding all of the incoming packets, the queue length is smaller than a predefined threshold value, no packet drop occurs. Otherwise, let the packet drop rate be D_k. Then the actual queue length after the packet drop is $Q_{k+1} = [Q_k + (1-D_k)P_k - L_k]^+$, which can also be written as $Q_{k+1} = [Q_k - D_k P_k + P_k - L_k]^+$. Since our goal is to minimize the packet drop rate and at the same time minimize the queue length, the performance function we try to minimize is defined as

$$J = \frac{1}{2}\sum_{k=1}^{N-1}\left(w_q Q_k^2 + w_d D_k^2\right),$$

where w_q and w_d are weighting coefficients for Q_k and D_k, respectively. w_q and w_d are introduced in the function to achieve different optimization performance. Different classes of multimedia flows have their own packet loss ratio and delay constraints. This differentiation is implemented via the different values of w_q and w_d. The decision of the values of w_q and w_d in real implementation depends on the packet loss ratio and delay (queue length in our equation) required for a particular class of flow.

The optimal packet drop rate at time interval k that minimizes the performance function is

$$D_k = -K_k Q_k + U_k v_{k+1}$$

where the values of K_k, U_k and v_{k+1} are obtained via solving the following equations [7].

$$K_k = -\left(P_k^2 S_{k+1} + w_d\right)^{-1} S_{k+1} D_k, \quad S_N = 0,$$

$$S_k = S_{k+1}\left(1 + P_k K_k\right) + w_q,$$

$$U_k = -\left(P_k^2 S_{k+1} + w_d\right)^{-1} P_k,$$

$$v_k = \left(1 + P_k K_k\right)^T v_{k+1} - \left(1 + P_k K_k\right)^T S_{k+1}(P_k - L_k),$$

$$v_N = 0.$$

After the packet drop rate is obtained, the new arriving packet is dropped at this percent by the router from the tail of the queue. The variables we use are all scalars and the computation overhead is low.

3. SIMULATION RESULTS

3.1 Drop Rates for Different Traffic Classes

In our simulation, we assume the values of w_q and w_d are known, and try to find out how the values affect the packet loss ratio and queue length. When there are several classes of traffic flows passing the router, different combinations of w_q and w_d can be chosen for different classes, and thus different QoS requirements of the different classes of traffic flows can be met. We select one queue scenario in our simulations. The total arrival rates of these packets are generated randomly between [0, 1 x 10^6] Bps, the total bandwidth available for this queue is 6 x 10^5 Bps, and the threshold value is set to 6 x 10^5 Bytes. The arriving packets indicate the total packets arriving at this queue at a

certain time interval, and queue occupancy represents the size of the packets in the queue at the same time interval.

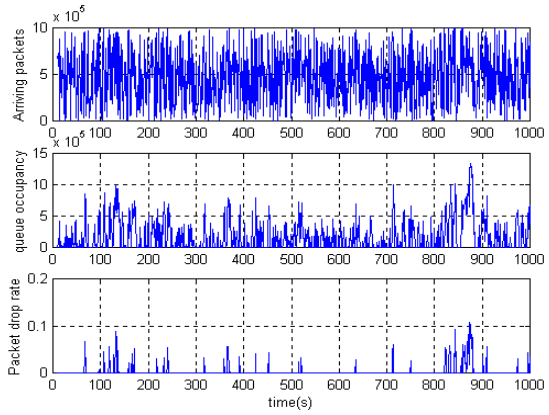

Figure 1. Queue occupancy and packet drop rate under $w_a = 1$ and $w_d = 10^{13}$.

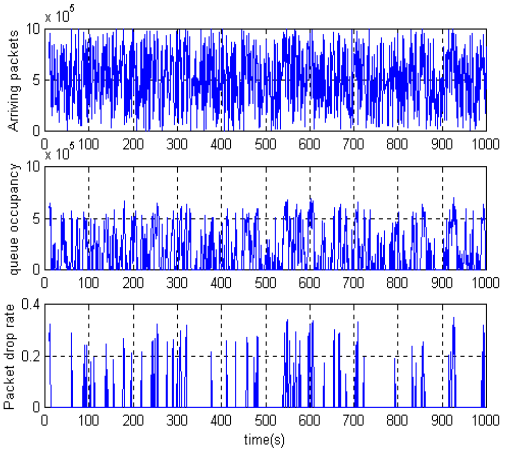

Figure 2. Queue occupancy and packet drop rate under $w_q = 1$ and $w_d = 10^{12}$.

In Figure 1, the queue occupancy, number of arriving packets and packet drop rate under $w_q = 1$ and $w_d = 10^{13}$ are shown. As can be seen from this figure, the packet drop rate is mostly kept below 10% with occasionally a little above that. The 10% packet loss rate is tolerable to most multimedia applications such as the video streams to provide an acceptable quality of presentations.

To illustrate how the combinations of w_q and w_d change the queue occupancy and packet drop rates, we ran the simulation with $w_q = 1$ and $w_d = 10^{12}$ as shown in Figure 2. In our experiments, we change only the value of w_d since we concerned more on the changes of packet drop rates. Compared with Figure 1, the queue occupancy in Figure 2 is smaller, while the packet drop rates are much larger (sometimes larger than 20%). This kind of high packet loss rate can occur at a very low bandwidth link. We can observe that a decreased w_d value is more suitable for those classes of traffic flows that can tolerate relatively larger packet drop rates but require smaller packet delays.

3.2. Comparison with the Fixed Drop Rate Mechanism

To illustrate the efficiency of our approach, we compared our approach (denoted as A) with the fixed drop rate approach (denoted as B). We also use a relatively large packet loss ratio that may happen during severe network congestion or in a low bandwidth link to examine how our approach performs under severe network congestion.

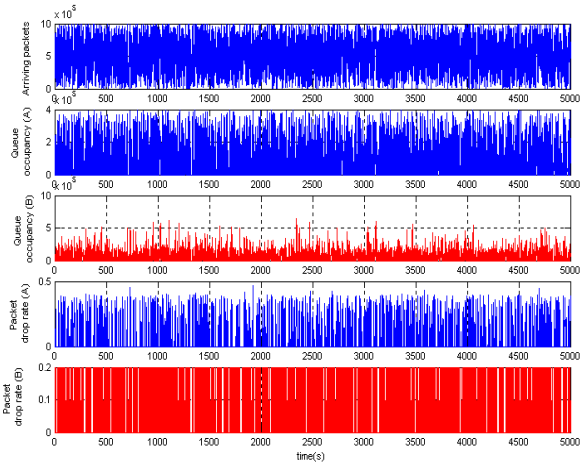

Figure 3. Comparison with fixed drop rate 20% in time interval [0, 5000].

In Figure 3, $w_q = 1$, $w_d = 10^6$, fixed drop rate = 0.2, drop queue length threshold = 2×10^5 Bytes and output rate = 6×10^5 Bps. The simulation was run within time interval [0, 5000]. As shown in Figure 3, the queue length in our approach changes more smoothly compared with the fixed drop rate mechanism. We do not have those peak queue length values that occur during the severe congestion situations. Also, the queue length and the packet drop rate obtained in our approach are both upper bounded. The queue length is maintained below 4×10^5 Bytes, and the packet drop rate is kept much below 50%. This has a practical meaning in implementation since the queue length in a router is actually limited. The router also provides a bound on the queue size because of its physical buffer size. The upper bound of the queue length we obtained can be dynamically adjusted according to the QoS requirements of different multimedia flows. The bounded queue length can also be used to provide an end-to-end delay bound for multimedia packets, especially for those interactive applications.

The two approaches are also compared under severe network congestion where the average packet drop rate is larger. In this simulation, $w_q = 1$, $w_d = 10^9$, fixed drop rate = 25%, packet arrival rates are generated randomly in a larger scope of [0, 2×10^6] Bps, and the output rate is increased to 10^6 Bps. As shown in Figure 4, the maximal packet drop rate can be maintained around 50%, and the queue occupancy is bounded below 10^6 Bytes. The smooth

queue occupancy of our approach is quite obvious. We observe that a decreased w_d value will provide a larger packet drop rate scope.

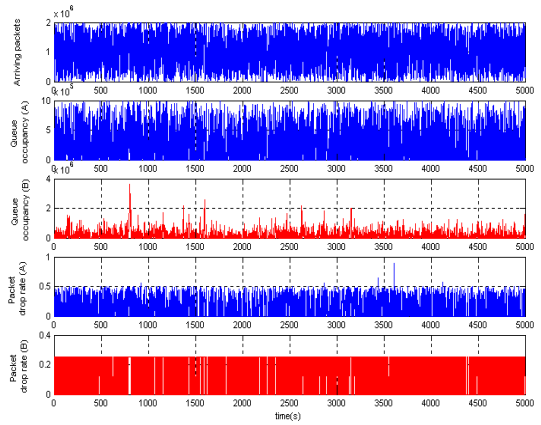

Figure 4. Comparison with fixed drop rate 25% in time interval [0, 5000] under severe congestion scenarios.

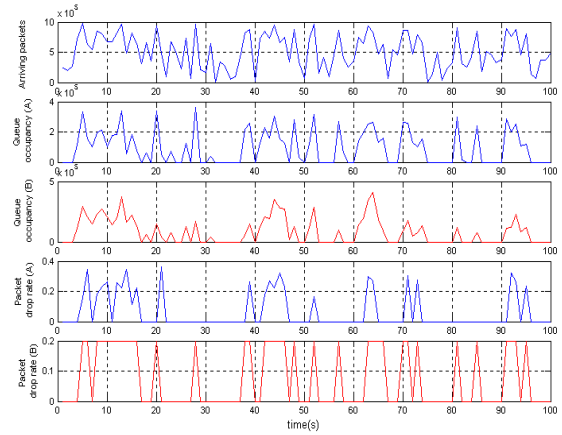

Figure 5. Comparison with fixed drop rate 20% in time interval [0, 100].

To give a better view of how our algorithm works, Figure 5 shows the simulation results during [1, 100] time intervals under the same parameters as in Figure 3. It can be seen from the figure, the packet dropping occurs more frequently in the fixed drop rate mechanism. Therefore, in the long run, it gives a better and more smoothly queue length, which helps to provide a more stable presentation quality for the multimedia applications in terms of playback delays and delay jitters. Even when the incoming packet rates change frequently and drastically during network congestion, our approach can still provide a relatively stable queue length and queue delay.

4. ACKNOWLEDGEMENT

This work was supported in part by Telecommunications & Information Technology Institute (IT2)/FIU, IT2 BA01.

5. CONCLUSIONS

In this paper, we present an active queue management approach that can be used in the router for congestion control. This approach can provide not only fairness between the TCP flows and the unresponsive UDP flows, but also QoS guarantee for the UDP flows of multimedia applications. The packet drop rates are adaptively determined according to the congestion status and the current queue occupancy. The packet drop rate is minimal and the resulting queue occupancy is also kept minimal. Comparisons are made with the fixed packet drop rate mechanism. Simulation results show that our approach can provide a more smoothly queue occupancy in a long run. The upper bounded queue occupancy is important to provide the class-based guaranteed delay services for real-time multimedia applications.

6. REFERENCES

[1] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, "REM: Active Queue Management," *IEEE Network*, pp. 48-51, May/June 2001.

[2] C. Diot, and J. L. Boudec, "Control of Best Effort Traffic," *IEEE Network*, pp. 14-15, May/June 2001.

[3] V. Firoiu, M., Borden, "A Study of Active Queue Management for Congestion Control," *Proc. Infocom*, pp. 397-413, 2000.

[4] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, 1(4), pp. 397-413, August 1993.

[5] P. Gevros, J. Crowcroft, P. Kirstein, and S. Bhatti, "Congestion Control Mechanisms and the Best Effort Service Model," *IEEE Network*, pp. 16-26, May/June 2001.

[6] P. Hurley, J. L. Boudec, P. Thiran, and M. Kara, "ABE: Providing a Low-Delay Service within Best Effort," *IEEE Network*, pp. 60-69, May/June 2001.

[7] F. Lewis and L. Syrmos. *Optimal Control.* John Wiley & Sons, INC., 1995.

[8] D. Lin and R. Morris, "Dynamics of Random Early Detection," *Proc. of SIGCOMM'97*, Cannes, France, pp. 127-137, September 1997.

[9] M.-L. Shyu, S.-C. Chen, and H. Luo, "Optimal Resource Utilization in Multimedia Transmission," *IEEE International Conference on Multimedia and Expo (ICME)*, Waseda University, Tokyo, Japan, pp. 880-883, August 22-25, 2001.

[10] M.-L. Shyu, S.-C. Chen, and H. Luo, "Self-Adjusted Network Transmission for Multimedia Data," *Third IEEE Conference on Information Technology: Coding and Computing (ITCC-2002)*, Las Vegas, Nevada, USA, pp. 128-133, April 8-10, 2002.