An Online Buffer Management Algorithm for QoS-Sensitive Multimedia Networks

Sungwook Kim and Sungchun Kim

ABSTRACT—In this letter, we propose a new online buffer management algorithm to simultaneously provide diverse multimedia traffic services and enhance network performance. Our online approach exhibits dynamic adaptability and responsiveness to the current traffic conditions in multimedia networks. This approach can provide high buffer utilization and thereby improve packet loss performance at the time of congestion.

Keywords—Active queue management, online decision, buffer management, resource reservation, QoS.

I. Introduction

Efficient network management requirements control decisions that are dynamically adjustable. However, at any point in time, the future rate of traffic arrival is generally not known, and there can be dramatic short-term variations in traffic patterns [1], [2]. These control decisions, therefore, have to be made in realtime. Online algorithms [3] are natural candidates for the design of efficient control schemes in QoS-sensitive multimedia networks. We propose a new online network management algorithm for QoS-sensitive multimedia networks which is designed to handle control decisions in an online manner.

Heterogeneous multimedia data is usually categorized into two classes according to the required QoS: class I (realtime) and class II (non-realtime). Class I data has a higher priority than class II data, so a multimedia network should take into account the prioritization among different multimedia traffic services [1], [2].

Our online decisions in each proposed mechanism are mutually dependent on each other under widely diversified network situations. In addition, our proposed algorithm concentrates on packet forwarding using appropriate queue management; no state information is needed in the core routers. Therefore, due to its stateless nature, it does not suffer from the scalability problem.

II. Proposed Online Buffer Algorithm

The class-based queue (CBQ) is a traffic management queuing algorithm. This algorithm sets different queues for different packet classes. Each class of traffic is assigned to a specific queue, each of which is guaranteed some portion of the total bandwidth of the router. Unlike the CBQ algorithm, our proposed algorithm is an online queuing algorithm for a shared buffer.

1. QoS Control Mechanism for Class I Services

The purpose of bandwidth reservation is to ensure some network capacity for class I traffic. For a given traffic load, there is an optimal amount of bandwidth which should be reserved, but this amount varies with the network traffic. To determine the optimal amount dynamically, we partition the time axis into equal intervals of length *unit_time*. Our proposed online algorithm adjusts the amount of reserved bandwidth (Res_B) based on realtime measurements during every *unit_time*.

To maintain the reserved bandwidth close to the optimal value, we define a traffic window which is used to keep the history of class I traffic requests (W_{class_I}). The traffic window is $[t_c - t_{class\_I}, t_c]$, where t_c is the current time, and t_{class_I} is the window length. This size can be adjusted in time steps equal to

Manuscript received May 8, 2007; revised June 14, 2007.

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2006-331-D00475), Sogang University under the 2006 New Faculty Support Program, and the Brain Korea 21 Project in 2007.

Sungwook Kim (phone: + 82 2 705 8932, email: swkim01@sogang.ac.kr) and Sungchun Kim (email: ksc@mail.sogang.ac.kr) are with the Department of Computer Science, Sogang University, Seoul, Rep. of Korea.

unit_time. It is increased or decreased if the call blocking probability (CBP) for new class I service requests is larger or smaller than its predefined target probability (P_{class_I}). The values of Res_B can be estimated as the sum of requested bandwidths by class I calls during the traffic window:

$$Res_B = \sum_{i \in W_{class\_I}} (B_i \times N_i) \,, \tag{1}$$

where N_i and B_i are the number of class I data requests and the corresponding bandwidth of data type *i*, respectively. Therefore, by using this traffic window, we can adjust the amount of reserved bandwidth (Res_B) at every *unit_time*, which is more responsive to changes in the network condition after the bandwidth has been reserved.

2. Congestion Control Mechanism for Class II Services

In contrast to class I traffic services, class II services do not need QoS guarantees. Therefore, instead of call-admission rules, congestion control mechanisms are required for class II traffic management in multimedia networks [4], [5]. Our proposed adaptive online congestion control mechanism differentiates between class I and class II traffic. When network congestion occurs, our mechanism attempts to provide a "better effort" service for class II traffic while maintaining the QoS of the call-admission controlled class I services and can achieve a high throughput and a low average delay.

With the aim of approximating optimal network performance while maintaining QoS guarantees, our proposed congestion control mechanism adjusts the AQM parameter values in an adaptive online fashion. The two buffer management parameters used are the queue range (Q_r) and the packet marking probability (M_p): Q_r is a threshold for traffic buffering, and M_p is assigned to drop class II data packets in a randomized manner.

Based on this flexible bandwidth sharing strategy between class I and class II traffic, our mechanism aims to ensure packet buffering until Q_r is reached. Therefore, Q_r is defined to be equal to the current Res_B. By inspecting the current reserved bandwidth, the Q_r value can be adaptively adjusted at every *unit_time*.

For adaptive network congestion management, we treat the M_p adjustment as an online decision problem by considering the current network traffic conditions. In this letter, M_p is also adaptively adjusted at every *unit_time* according to the current queue situations. The ideal situation for well-balanced network performance is for the current queue length (L) to be the same as Q_r. Therefore, L is used as the main indicator for determining M_p. In our proposed mechanism, three system parameters are used to determine M_p: the maximum queue length (ML), L, and Q_r. Based on these parameters, we define two packet marking probabilities (M_{p_1} and M_{p_2}) that are more adaptable to traffic fluctuations.

$$M_{p\_1} = \frac{L_r}{ML} \quad \text{and} \quad M_{p\_2} = \frac{L - Q_r}{ML - Q_r}. \tag{2}$$

Here, L is used as the main factor for determining M_{p_1}, whereas we consider Q_r when determining M_{p_2}. This allows M_{p_2} to be changed rapidly in the presence of heavy traffic congestion.

Since the future is not known exactly, we assume that the traffic pattern during the recent time interval [t_c – *unit_time*, t_c] reflects the current traffic situation. Therefore, our algorithm monitors the packet queuing rate (I_{p_r}) in the recent time interval, which is defined as the rate at which packets are being queued in the traffic buffer. Based on the recent traffic condition, we decide whether packet overflow (or underflow) will occur; that is, whether the number of incoming bits (I_b) to the queue will be larger (lower) than the outgoing bits (O_b). When a packet overflow occurs ($I_{p\_r} > 0$), L increases, and an underflow situation results in L decreasing.

In the presence of traffic congestion, our online congestion control mechanism drops packets by reducing M_p when packets are queued for the interval [t_c, t_c + *unit_time*]. Dropping packets provides feedback information to source nodes on the congestion level of the gateways through the path. To make control decisions more responsive to changes in the current traffic, we categorize the status of the buffers into four types based on L and I_{p_r}: congestion, potential-congestion, potentially-safe, and safe.

When L is greater than the total buffer size (T), this indicates that the current traffic load in the network is very high and that the buffer does not have sufficient space for the incoming packets. Therefore, the buffer status is defined as congestion, and all arriving class II data packets should be dropped ($M_p = 1$). When L is less than Q_r ($0 < L < Q_r$), there are two sub-cases based on the recent I_{p_r}. If the available buffer space ($T - Q_r$) is sufficient to ensure current traffic overflow ($(T - Q_r) > I_{p\_r}$), the buffer status is defined as *safe*, whereby the network situation is considered congestion free and no arriving packets are dropped ($M_p = 0$). If $((T - Q_r) < I_{p\_r})$, the buffer status is defined as potential- safe, whereby the current available buffer space is not sufficient to support the current traffic overflow. Therefore, we set $M_p = M_{p\_1}$ to reduce the current traffic overhead.

When L is greater than Q_r, but less than T ($Q_r < L < T$), there are also two sub-cases based on the recent traffic condition (I_{p_r}). If an overflow occurs ($I_{p\_r} > 0$) under a situation of $Q_r < L < T$, the buffer status is defined as potential-congestion, whereby the total available buffer space can be assumed to reach a critical low point in the near future. Therefore, we set $M_p = M_{p\_2}$ to catch up with the traffic congestion. Even though under the situation of $Q_r < L < T$ the amount of over-buffered traffic ($L - Q_r$) can be controlled by the current underflow situation ($(O_b - I_b) > (L - Q_r)$), we assume that the current network congestion will be resolved by the occurrence of traffic underflow ($I_{p\_r} < 0$) in a future time interval. Therefore, the buffer status is defined as safe, and we set $M_p = 0$.

III. Analysis and Conclusion

Using a simulation model, we compare the performance of our proposed online bandwidth management framework with two existing active queue management schemes: the random early detection (RED) scheme [4] and the BLUE scheme [5].

In our simulation model, we make the following assumption: new call requests arrive according to a Poisson process with call arrival rate from 0 to 3.0 calls/s. Based on this assumption, *unit_time* in our simulation model is one second.

Figures 1 to 3 show the simulation results for traffic including both class I and class II services. The curves in these figures show that our scheme (ORANGE) exhibits superior performance to RED [4] and BLUE [5] schemes in terms of the average delay, packet loss rate, and call blocking probability under various traffic conditions. Figure 4 shows the simulation result of call blocking probability for traffic of only higher-priority class I service, also under various traffic conditions.

Generally, our proposed scheme exhibits superior performance to the existing schemes [4], [5] under light to heavy traffic loads. Specifically, our scheme balances the performance between contradictory requirements while other schemes cannot offer such an attractive trade-off. The main novelty of our proposed scheme is to respond to feedback from realtime estimation. Due to traffic uncertainty, our online approach provides adaptability, flexibility, and responsiveness to current traffic conditions in multimedia networks.

References

[1] S.W. Kim and P.K. Varshney, "An Adaptive Bandwidth Allocation Algorithm for QoS Guaranteed Multimedia Networks," *Computer Comm.,* vol. 28, Oct. 2005, pp. 1959-1969.

[2] S.W. Kim and P.K. Varshney, "An Integrated Adaptive Bandwidth Management Framework for QoS Sensitive Multimedia Cellular Networks," *IEEE Trans. Vehicular Technology*, May 2004, pp. 835-846.

[3] Y. Azar, *Online Algorithms: The State of the Art*, Springer, 1998.

[4] J. Orozco, D. Ros, J. Incera, and R. Cartas, "A Simulation Study of the Adaptive RIO (A-RIO) Queue Management Algorithm," *Computer Comm.*, vol. 28, no. 3, Feb. 2005, pp. 300-312.

[5] B. Okuroglu and S. Oktug, "BIO: An Alternative to RIO," *SPIE ITCom: Int'l Symp. Convergence of IT and Comm.*, Denver, USA, Aug. 2001, pp. 4524-4532.

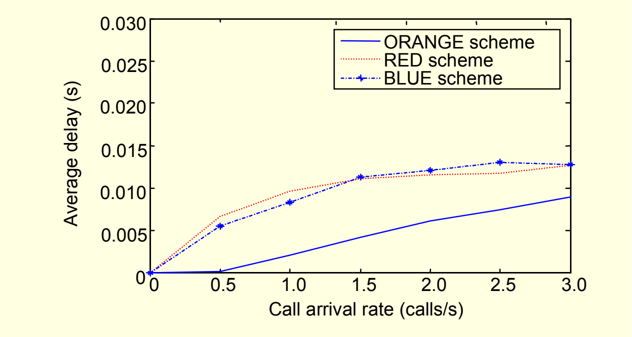

Fig. 1. Comparison of average delay.

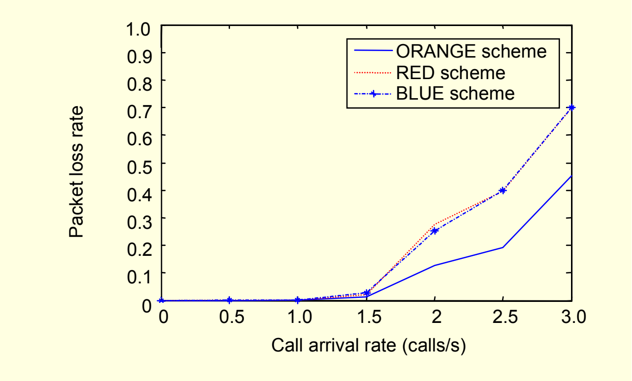

Fig. 2. Comparison of packet loss rate.

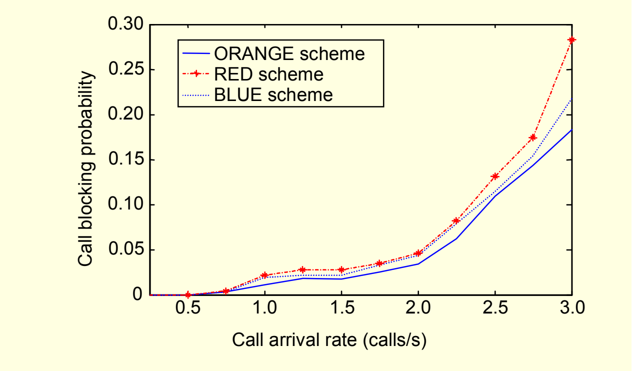

Fig. 3. Comparison of call blocking rate for mixed traffic.

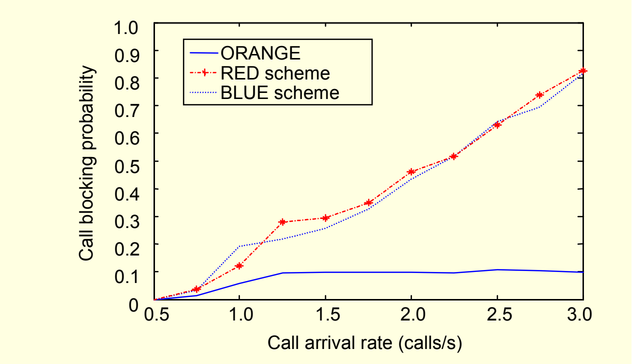

Fig. 4. Comparison of call blocking probability for class I traffic.