

An Efficient Dynamic Threshold Buffer Allocation Scheme for the Future Internet

D.B.Pillai¹, G.Ojong², Dr.S.S.Xulu³

Department Of Computer Science, University of Zululand

Private Bag X1001, KwaDlangezwa, 3886

1. Email: divyabp@gmail.com, 2. Email: gojong@pan.uzulu.ac.za

3. Email: sxxulu@pan.uzulu.ac.za

Abstract – The current Internet uses a single queue per output port to buffer packets destined for that port. This often causes congestion leading to packet loss and delay. Real-time applications are delay and loss sensitive. There is therefore a need to develop a buffer management system that will effectively accommodate both real-time and non-real-time applications. In this paper, we propose a dynamic threshold buffer management scheme. This scheme uses complete sharing with virtual partitioning. Pre-emption is used to minimise congestion. Out of profile high priority packets are dropped during pre-emption period. To add fairness to the scheme, low priority packets are guaranteed a minimum buffer space.

We simulated the scheme and found that DTBAS performs well in terms of packet loss and buffer utilisation.

Keywords – Congestion control, buffer management, buffer allocation, quality of service

I. INTRODUCTION

The number of new applications that are being developed for the Internet is increasing tremendously. The present trend is towards the deployment of real-time multimedia applications, in addition to the non-real-time applications that the Internet has been transmitting since its inception. Multimedia applications such as voice, video, audio-video on-demand, and video conferencing, demand not only high bandwidth but also real-time delay constraint. In order to properly transmit these real-time multimedia applications, a variety of resources have to be made available. Bandwidth and buffers are the main network resources that need to be considered in the transmission of these applications (flows). Many real-time applications require large amounts of buffers to store packets effectively.

Efforts have been made to introduce quality of service (QoS) on the Internet and researchers have proposed new architectures for the Internet. These QoS architectures are designed to make network components QoS aware. The Internet Engineering Task force has been the driving force behind finding a suitable QoS architecture. They have proposed the Integrated Services (IntServ) [1], and the Differentiated Services (DiffServ) [2]. In addition, the Rainbow Services [3] has also been proposed. A common feature in all these proposed QoS models is that, there is the need to allocate resources, either implicitly or explicitly. Appropriate resource allocation is one way of providing QoS in a network.

In a network, congestion is created if bandwidth and buffers are not properly managed within the hosts or at the routers of the network. A communication link can become

congested if the amount of traffic introduced on the link exceeds its capacity. This excess traffic causes queuing delays to increase rapidly as buffers fill up, and in extreme cases can cause the buffers to overflow, losing packets. When this occurs, the network will not be able to provide the guarantee made to some applications.

New Internet applications such as real-time multimedia applications do not adapt well to congestion. During the transmission of these applications, buffers become depleted very fast, causing many packets to drop. It is important to avoid high packet loss rates because when a packet is dropped before it reaches the destination, all of the resources it has consumed in transit, are wasted. Therefore, appropriate buffer management mechanisms are required in order to alleviate the congestion at the buffers.

Buffer management techniques involve scheduling of resources to applications, packet dropping and buffer allocation. Scheduling is used to move packets out of the buffers in a specified order and to manage delay within the buffers. Packet dropping is used to maintain the buffer size, in order to efficiently handle both real-time and non-real time packets. Many buffer management schemes that have been proposed have focused either on packet dropping [4,5,6], packet scheduling [7] or both [8]. Buffer allocation, as an independent mechanism, has not been properly investigated. Buffer allocation is an aspect of buffer management that focuses on managing buffers by allocating buffer volume (logical unit of space) to each application class, so as to reduce packet loss due to lack of buffer space.

Previously proposed buffer management schemes [4,5,6,8], are not capable of efficiently managing buffers when real-time multimedia applications are involved, as many of these schemes were developed with non-real-time applications in mind. Hence, we propose DTBAS, a dynamic threshold buffer allocation scheme, to efficiently manage buffers in the future Internet. Though real-time and non-real-time applications share the same buffer space, in the DTBAS, they are logically separated. This sharing of common buffer space is done by dynamically varying buffer thresholds so as to reduce packet loss rate of high priority packets, as well as increasing the utilization of the buffer space. Additionally, the scheme uses the concept of marked packets to prevent the dropping of useful packets when congestion is severe.

This paper therefore proposes a class-based QoS control scheme that allows the dynamic allocation of buffers for various types of flows. It also manages packet loss rates within the buffers to efficiently transmit real-time and non-real-time applications over the Internet.

The remainder of this paper is organised as follows: Section II looks at some previously conducted work in this field. Section III describes the proposed scheme. We

compare the performance of our scheme with two previously proposed schemes and present our results in Section IV. Finally the paper concludes in Section V.

II. RELATED WORK

Considerable research effort is currently underway to develop a suitable buffer management scheme that will accommodate high bandwidth applications on the Internet, as well as reduce large amounts of packet loss during congestion [9,10,11,12,13]. Work has also been done on managing both bandwidth and buffer to provide QoS [14,15]. There is also ongoing research in the wireless field on the allocation of buffers [16,17,18]. In this section we review some of the existing buffer management schemes. We focus mainly on buffer allocation strategies in the output buffers of routers on the Internet.

The class-based QoS control scheme by Minami *et al.* [10], uses a single buffer that is shared among all the output ports. This scheme uses a hierarchical structure in the order of ports, classes and flows. In this scheme, the total buffer size is shared by output ports according to the ratio of each output link speed. The basic volume is allocated per flow. This increases the complexity of the scheme. Also delay is introduced due to per flow calculations.

Yaprak *et al.* [12] proposed a dynamic buffer allocation scheme that uses a shared architecture with virtual partition among output ports. Instead of using pre-emption, they created a shared buffer pool where excess of buffer space from any logical queue goes into. The problem with this technique is that the shared buffer pool may be monopolised by non-real-time applications thereby reducing the QoS of real-time applications. They neither considered using pre-emption nor the marking of packets.

Complete Sharing using Virtual Partition (CSVP) scheme by Wu and Mark [11] was introduced to manage multi-traffic flows in ATM networks. In this scheme, if there are M number of users and the available buffer space is B , the B buffer spaces are virtually partitioned into m segments corresponding to the M number of users. A newly arriving cell is admitted by pushing out a different type of cell that is borrowing that cell's buffer volume. The authors of the scheme did not consider the possibility of marking packets as either IN-profile or OUT-profile.

The scheme proposed by Garcia-Marcias *et al.* [16], shares buffer in a linear manner. In this scheme class packets enter into the output buffer and each of these class packets are allocated buffer volume. This scheme completely partitions its buffer space among the various classes. This reduces the efficiency of the scheme. Furthermore, pre-emption of low priority packets was not considered.

The goal of the authors of [18] was to develop a QoS model to support multimedia applications over Cellular IP by using class-based service without reservations. They used per-class buffers with finite sizes. Using static buffers, it is not possible to provide excess buffer to other applications especially real-time applications. Pre-emption was not used.

This paper focuses on the use of mainly buffer allocation to minimize both delay and loss rate. Unfortunately buffer allocation alone cannot totally reduce delay in transmission. It is the proper combination of buffer allocation, packet dropping and scheduling mechanisms that would produce an optimal strategy. This study uses simple

packet dropping and scheduling mechanisms so that the effect of the buffer allocation strategy will be largely felt.

III. PROPOSED ALLOCATION SCHEME

The Dynamic Threshold Buffer Allocation Scheme (DTBAS) is a class-based scheme, which is structured into a hierarchy of ports and classes. The scheme uses complete sharing with virtual partitioning plus pre-emption, to allocate packets to the buffer. The total buffer volume is virtually partitioned into a number of classes. A threshold-based packet discarding mechanism is used. Applications are grouped into two classes. Each class can access the other class' buffer volume when a class' assigned buffer volume is depleted. Classes use their upper threshold as a pointer to indicate the instantaneous total buffer space available to each of them. This threshold may vary depending on network conditions.

A router's output buffer consists of ports with multiple queues per port. Each queue is assigned to a class of traffic. Hence, ports consist of classes and classes consist of packets. We do not group class packets into flows. Packets in a class are treated as a single group, just as in DiffServ [2]. In this study only two classes are used (Class 1 and Class 2). Class 1 has a higher priority than Class 2; Class 1 is for real-time applications while class 2 is for non-real-time applications. Packets of Class 1 flows are marked as either IN-profile or OUT-profile. During congestion period, OUT-profile packets are dropped first before IN-profile packets, so as to minimise the loss rate of useful IN-profile packets. High-priority Class 1 packets can pre-empt low priority Class 2 packets.

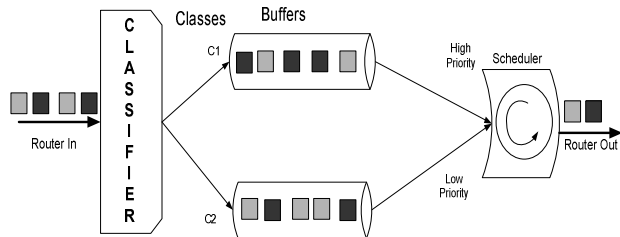


Figure 1: Single port architectural model of DTBAS

Figure 1 depicts the architecture of a single port in the router. It consists of a classifier, two logical buffers (C1 and C2) and a scheduler. Arriving packets are classified as either class 1 and sent to C1 logical buffer or as class 2 and sent to C2. Packets of each logical buffer are then scheduled; with packets in C1 given preferential treatment. This scheme uses virtual partitioning to separate the buffer space for packets of each class. We used Drop tail with dynamic threshold to drop the packets of Class 1 and Class 2 flows.

When a Class 1 high-priority packet enters its designated buffer space and finds it full, it looks for any available space in Class 2's logical buffer, to borrow. If there is an available space within Class 2's logical buffer, packets of Class 1 will then use this space and the upper threshold of Class 1 is increased accordingly. The space can be borrowed until the minimum threshold of class 2 is reached. During this period, Class 1 starts dropping its OUT-profile packets. When the minimum threshold of Class

2 is reached, all arriving class 1 packets will be dropped. If Class 1 buffer is free, Class 2 packets could also use it.

IV. SIMULATION MODEL AND RESULTS

To evaluate the performance of our model we conducted a simulation. The simulation design consisted of two hosts, a router and a sink, as shown in Figure 2. Each host generates packets and sends them to the router, where they are classified, queued and then scheduled out to the sink. We used a two-service-class model; Class 1 (for real-time applications) and Class 2 (for non-real-time applications).

Host 1 generates packets of Class 1 while Host 2 generates packets of Class 2. The generated packets are sent to the router where they are classified and stored in either Q1 or Q2. Class 1 packets are stored in Q1 while Class 2 packets are stored in Q2. A certain amount of buffer volume is assigned to each class. The total buffer space is partitioned as follows: x for Class 1 and y for Class 2.

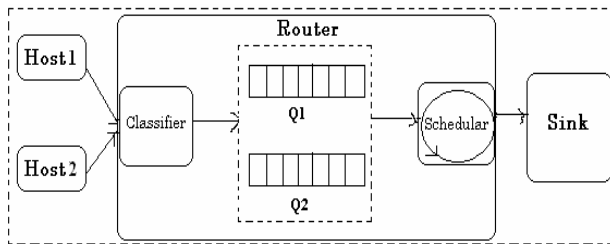


Figure 2: Simulation Model

We conducted several experiments during the simulation. The goal of conducting the experiments was to observe and analyse the behaviour of the DTBAS scheme in managing buffers for both real-time and non-real-time applications. Packet Loss Rate (PLR) and buffer utilisation were used as performance metrics; against simulation time. Experiments were also conducted to compare the packet loss rate and buffer utilisation of the DTBAS scheme, with previously proposed schemes such as Complete Partitioning (CP) and Complete Sharing (CS) buffer allocation schemes. All the experiments used two traffic classes; Class 1 and Class 2. Class 1 (for real-time application) is represented here as Class 1, while Class 2 (for non-real-time application) is represented as Class 2. Drop Tail (DT) with dynamic threshold packet discarding mechanism was used to discard packets during congestion. The schemes were then subjected to Poisson traffic, a widely used traffic model for simulating Internet traffic. This traffic type has also been used by most of the previously proposed schemes.

The scheduling mechanism used in this scheme is the Weighted Round Robin mechanism. Each class' queue is assigned a certain weight, and Class 1 with the highest weight sends out more packets than Class 2.

A. Packet Loss Rate in DTBAS

The purpose of the experiment was to compare the packet loss rate (PLR) of Class 1 and Class 2 flows in the DTBAS. In this scheme, Class 1 and Class 2 were assigned equal buffer volume. Each class' upper threshold was initially made equal to its buffer volume. Figure 3 shows how the PLR varies with time for Class 1 and Class 2 flows.

From Figure 3, it can be noticed that Class 2 packets are dropped far earlier than packets of Class 1. It can also be seen that after 1000 seconds, the packet loss rate of Class 1

is always far lower than that of Class 2. This is due to packets of Class 1 being able to pre-empt Class 2 packets, at full buffer volume. At this time the threshold of Class 2 is decreased until a minimum buffer volume is attained. Correspondingly, the upper threshold of Class 1 is increased to include the buffer volume gained from Class 2. Thus, using a dynamic threshold-based sharing with pre-emption and marking, helps in reducing packet loss rate of Class 1 flows.

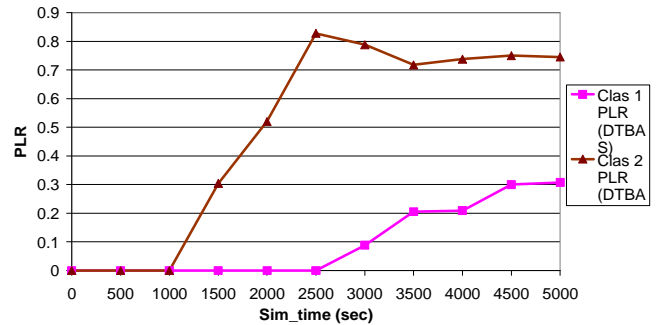


Figure 3: Per-class PLR in DTBAS

B. PLR of DTBAS compared with those of CP and CS buffer allocation schemes

This test was conducted to compare the packet loss rate of DTBAS with those of the CP and CS schemes. Each of the schemes was subjected to the same packet arriving and departing rate. Figure 4 shows how PLR varies with time. Each scheme's Class 1 flow is depicted here.

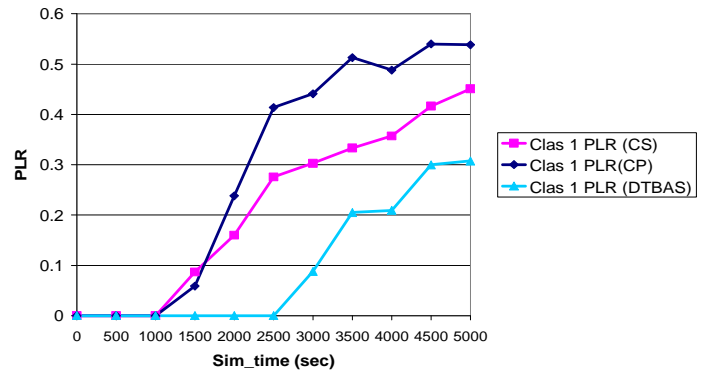


Figure 4: Class 1 PLR of CP, CS and DTBAS schemes

From Figure 4, it can be observed that it takes a longer time before Class 1 packets in the DTBAS start dropping, when compared to the CP and CS schemes. It can also be seen that the packet loss rate of Class 1 packets of the DTBAS is far lower than those of the CP and CS schemes. This is due to the fact that in the DTBAS, at full buffer volume, high priority packets of Class 1 pre-empt low priority packets of Class 2. Also, the system starts dropping arriving OUT-packets at the output queue, creating more space for IN-packets to be stored.

C. Buffer utilisation of DTBAS compared with those of CP and CS buffer allocation schemes

In this experiment, the performance of DTBAS, in terms of efficient buffer utilisation, was compared with those of the CP and CS schemes. The buffer volume

assigned to each class was the same in all three schemes. Figure 5 shows how buffer utilisation varies with time.

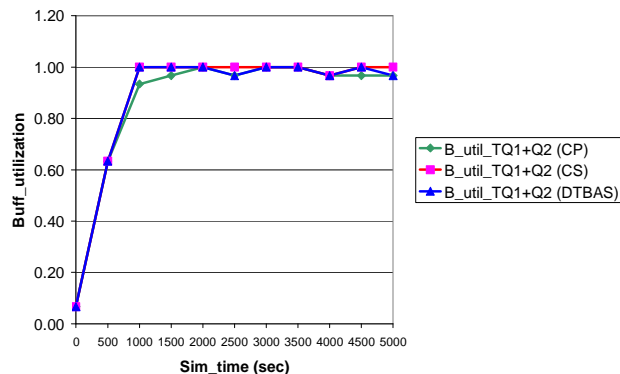


Figure 5: Buffer utilisation for CP, CS and DTBAS

It can be observed from Figure 5, that the buffer utilisation of DTBAS is comparable to that of the CS scheme. This is because of the sharing of buffer space among packets of the two classes, in both schemes, especially during periods of congestion. DTBAS therefore utilises buffer efficiently just like the CS scheme.

V. CONCLUSION

Congestion on the Internet results in long queues, forcing packets to be dropped. This affects the quality of real-time multimedia applications transmitted on the Internet. Many mechanisms to alleviate this congestion have been proposed but very few have dwelt on the use of buffer allocation strategies to reduce the effect. In this study, a Dynamic Threshold Buffer Allocation Scheme (DTBAS) has been proposed. In DTBAS, applications are classified as either real-time or non-real-time. Real-time Class 1 packets can pre-empt non-real-time Class 2 packets during periods of congestion. Class 1 packets are marked as IN-profile or OUT-profile. OUT-profile packets of Class 1 are dropped first.

Simulation results show that the DTBAS scheme performed best in terms of packet loss rate (PLR) than the CS and the CP schemes. It was also found that the buffer utilisation of the DTBAS is comparable to that of the CS scheme. These results make the DTBAS one of the best schemes to be considered for the future Internet.

Future work on DTBAS could use delay as a performance metric. The DTBAS scheme's performance could further be investigated using a combined packet dropping and scheduling mechanism. DTBAS' performance could also be compared with Cisco's low latency queuing (LLQ) scheme.

REFERENCES

- [1]. R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", *IETF RFC 1633*, June 1994.
- [2]. S. Blake, D. Black, M. Calson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services", *IETF RFC 2475*, December 1998.
- [3]. G. Ojong and F. Takawira, "Rainbow Services: A New Architecture for QoS Provisioning in the Future Internet", *Proceedings of SATNAC '03*, September 2003.

- [4]. B. Branden, D. Clark, B. Davie, S. Floyd, G. Minshall, L. Peterson, K. K. Ramakrishnan, S. Shenker, and L. Zhang, "Recommendation on Queue Management and Congestion Avoidance in the Internet", *IETF RFC 2309*, April 1998.
- [5]. S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, vol. 1, Issue 4, pp. 397–413, 1993.
- [6]. S. Athuraliya, V. Li, S. Low, and Q. Yin, "REM: Active queue management," in *IEEE Network*, vol. 15, pp. 48–53, May/June 2001.
- [7]. C. Semeria, "Supporting Differentiated Service Classes: Queue Scheduling Disciplines", *Juniper Network*, 2001.
- [8]. D. Clark and W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service", *IEEE/ACM Transactions on Networking*, vol. 6, Issue 4, pp. 362–373, August 1998.
- [9]. L. Chuang and Y. Li, "Dynamic Partial Buffer Sharing scheme: Proportional Packet Loss Rate", *Proceedings of the International Conference on Communication Technology*, Beijing, China, pp. 255–258, April 2003.
- [10]. K. Minami, H. Tode, and K. Murakami, "Class-Based QoS Control Scheme by Flow Management in the Internet Router", *Proceedings of the 25th Annual IEEE Conference on Local Computer Networks*, pp. 58–65, 2000.
- [11]. G. L. Wu and J. W. Mark, "A Buffer allocation Scheme for ATM Networks: Complete Sharing Based on Virtual Partition", *IEEE/ACM Transactions on Networking*, vol. 3, Issue 6, December 1995.
- [12]. E. Yaprak, A. T. Chronopoulos, K. Psarris, and Y. Xiao, "Dynamic buffer allocation in an ATM switch", *Computer Networks*, vol.31, pp. 1927–1933, 1999.
- [13]. Awan, I., Ahmad, S., and Ahmad, B., "Performance analysis of multimedia based web traffic with QoS constraint", *Journal of Computer and System Sciences*, vol. 74, pp. 232–242, 2008.
- [14]. Y. Zhou and H. Sethu, "On achieving fairness in the joint allocation of buffer and bandwidth resources: Principles and algorithms", *Computer Networks*, vol.50, Issue13, pp. 2239–2254, 15 September 2006.
- [15]. S. Jordan, K. Jogi, C. Shi, and L. Sidhu, "The Variation of Optimal Bandwidth and Buffer Allocation With the Number of Sources", *IEEE/ACM Transactions on Networking*, vol. 12, Issue 6, December 2004.
- [16]. J. A. García-Macías, F. Rousseau, G. Berger-Sabbatel, L. Toumi, and A. Duda, "Quality of Service and Mobility for the Wireless Internet", *Proceedings of the first ACM workshop on Wireless Mobile Internet*, pp. 34–42, July 2001.
- [17]. A. Malla, M. El-Kadi, and P. Todorova, "A Fair Resource Allocation Protocol for Multimedia Wireless Networks", *Proceedings of International Conference on Parallel Processing*, pp. 437, 2001.
- [18]. E. Sidahmed and H. Hesham, "A QoS Based Model for Supporting Multimedia Applications over Cellular IP", *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005.

D. Pillai is currently pursuing her MSc. Degree in the Department of Computer Science at the University of Zululand. Her area of interest is on Quality of Service provisioning on the Internet.