DEADLINE CREDIT SCHEDULING POLICY FOR PRECORDED SOURCES*

Zoe Antoniou
Nokia Research Center
3 Burlington Woods Drive
Burlington, MA, 01803, USA
zoe.antoniou@nokia.com

Ioannis Stavrakakis
Department of Informatics
University of Athens
Athens, 15784, Greece
istavrak@di.uoa.gr

Abstract

In this paper a scheduling policy is proposed, called Deadline Credit (DC) policy, which focuses on the efficient scheduling of prerecorded continuous media traffic. Its objectives are (a) to utilise all (if possible) of the available capacity by allocating bandwidth dynamically *to a number of data streams, (b) to distribute the available bandwidth* fairly *between all competing data streams and (c) to maximise the number of Application Data Units (ADUs) that are served by always giving priority to the one with the shortest time to expiration. Numerical results are presented and the induced ADUs loss rates under the proposed policy are compared against those of some alternative schemes.*

1 Introduction

The transmission of continuous media traffic is an inherently difficult problem due to the time sensitive nature and the traffic variability of these applications [1], [2].

Consider, for instance, the transmission of video applications in which it is required that frames (images) be transmitted at a certain rate in order to guarantee no starvation or overflow at the receiving end. Bits generated over a frame form a logical block of information (Application Data Unit or ADU) which is central to both the traffic generation process at the transmitter and the reconstruction process at the receiver. Such a role may be played by a smaller encoding unit than the frame such as a macroblock [3]. Since frames are typically generated at a constant frame rate the nature of such an application is periodic.

ADUs can be mapped into a collection of network date units or ATM cells. In order to guarantee a timely reconstruction at the receiver, complete frames should be delivered at a constant display rate. This implies that all cells of an ADU should be delivered in time for the reconstruction process in order to avoid either starvation or buffer overflow at the receiving end. When network resources are reserved at peak demand, all cells of the ADU will be delivered on time, leading to a timely and error-free reconstruction at This is clearly the case when resources are over-allocated to a number of applications to increase network utilisation.

The requirement for timely delivery of information associated with such continuous media traffic suggests that deadlines could be defined and deadline-driven policies be designed. If the nature of an application requires that a complete ADU should be received before it can be processed by the receiver, then it would be reasonable to associate a common deadline with all the cells in the same data unit and develop scheduling policies that aim to maximise the number of complete data units that are delivered by their respective due date. This approach is different from many traditional approaches in which every cell is assigned its own deadline and the scheduling policy typically attempts to maximise the number of individual cells transmitted by their due date [4], [5].

Continuous media traffic may be real time or prerecorded such as Video on Demand (VoD). In the prerecorded case, which is the focus of this paper, there can be various alternatives to the efficient transmission of the data since the information is available *a priori*, as opposed to the real time traffic where real time processing and transmission is the only approach. In either case, they both have a periodic *consumption* rate once the application starts running.

The proposed policy, called Deadline Credit (DC) policy, focuses on the efficient scheduling of prerecorded continuous media traffic. Its first objective is to utilise all (if possible) of the available capacity by allocating bandwidth *dynamically* to a number of (continuous media) data streams. Secondly, it aims to distribute the available bandwidth *fairly* between all competing data streams. Finally, it aims to maximise the number of ADUs that are served by always giving priority to the one with the *shortest time to expiration*. In order to achieve this the DC policy uses all of

*This work is part of Zoe Antoniou's Ph.D. research while with Northeastern University and has been supported by the National Science Foundation under Grant NCR 9628116.

the available bandwidth to (a) serve the ADUs with upcoming deadlines and (b) send ADUs in advance. This deadline credit can be *consumed* at a later time when the demand for resources may increase. The DC policy aims to build up the deadline credits of all competing streams fairly which ensures that no stream can monopolise the available bandwidth at the expense of other applications.

The problem of scheduling and transmitting prerecorded traffic and, in particular, video has been addressed in [3], [6] where a policy called Constant-Rate Transmission and Transport (CRTT) was proposed. According to CRTT, a number of frames (ADUs) are sent in advance (prior to the commencement of playback). This provides an initial "build up" and enables the remaining frames to be transmitted at a constant rate during playback. The amount of "build up" and the rate of transmission are computed given the characteristics of the movie which are known a priori and the amount of buffering available at the receiving end. However, the issue of multiplexing a number of streams carrying prerecorded traffic or how the losses will be distributed between streams sharing the same link for high network utilisation is not directly addressed as it is for the policy proposed here. In addition, the allocation of network resources is static and is determined once at the beginning of the session, as opposed to the dynamic allocation performed by the DC policy based on the varying network resource requirements from all users. The CRTT policy is rate based and deadlines are not explicitly defined as opposed to the DC policy.

2 The DC policy

In this paper a source node with one outgoing link is considered. The source node consists of a mass storage device, a file server and the scheduler/server. The file server retrieves, upon request, streams of continuous media traffic. These streams are multiplexed and transmitted into the ATM network by the scheduler/server. The system is assumed to be slotted; a cell requires a fixed amount of time for transmission referred to as a slot. Let j denote a data stream (application); $0 \leq j \leq N-1$. A variable with superscript j represents a quantity associated with stream j. Also, let P_i^j denote the i^{th} ADU from stream j with $\hat{P}_i^j$ representing its length in cells. For real time traffic a new ADU is assumed to be generated periodically every T^j slots, where T^j denotes the period of the generation process of the application measured in slots. So for real time traffic the number of ADUs available to the server at any time is limited by the generation process rate of the application. For prerecorded traffic though, all of the ADUs of the session are available prior to the commencement of transmission.

The application is assumed to have a periodic reconstruction rate; an ADU has to be available to the receiver every T^j slots for processing (T^j is assumed to be both the generation and the reconstruction periodic interval of the application). In order to meet this constraint, an ADU can tolerate a maximum delay (queuing plus transmission) at the source node referred to as the deadline. For real time traffic this deadline is defined as D^j slots from the instant marking the beginning of the corresponding periodic interval. Similarly for the prerecorded traffic an ADU must be served within D^j slots of the beginning of the corresponding periodic interval. For instance, the i^{th} ADU corresponds to the i^{th} periodic interval and its deadline is defined as D^j slots from the instant marking the beginning of the i^{th} periodic interval and so on. An assumption is made in this paper that the length of any ADU will be less than or equal to its default deadline. Also, it is assumed that for any stream j: $D^j \leq T^j$. The QoS metric considered here is the ADU loss probability. An ADU is assumed to be lost if any one of its cells is lost.

The DC policy is *work conserving* in the sense that it will serve as long as there are ADUs available for service and buffer space is available at the receiver. The proposed scheduling algorithm determines when the next stream is to be selected for service, which stream is to be selected and whether an ADU from the selected stream is to be served or dropped. A set of metrics is used in making the above decisions, (a) the deadline credit (with an associated counter, D_CR^j), the losses (with an associated counter, L_CR^j),and the priority index (with an associated counter, P_CR^j), as well as the buffer availability of the receiving buffer. The buffer availability is deduced from the deadline credit counter as it is explained later.

It is assumed that key events occur and actions are taken instantaneously at the end of slots. If during a slot either the last cell of an ADU or no cell is being served, the server will be idle at the end of this slot and this is referred to as a **decision slot**. At decision slots certain actions are taken leading to the selection of a stream to be served next. For the selected stream this decision slot becomes an **examination slot**; ADUs of stream j can be scheduled for service or dropped only at examination slots of stream j. The losses counter of a stream is updated only at the examination slots of this stream as it will be discussed later.

The **Additional Delay Tolerance** (ADT) of an ADU of stream j is defined to be equal to the amount of remaining time until the ADU expires reduced by its default deadline D^j. That is if ADT is equal to d_cr

for an ADU of stream j, this ADU will expire after $d\_cr + D^j$ slots. Finally, the ADU **at the head of stream** j is defined to be the ADU of stream j which has the earliest due date among all ADUs of stream j not served or dropped yet. As it will become clear later, expired ADUs remain in the system until the next examination slot for the corresponding stream at which time they are dropped.

The purpose of the deadline credit counter is to keep track of the ADT of the ADU at the head of the corresponding stream. At the same time, the deadline credit counter indicates how far ahead or behind schedule is the service of the ADUs of the corresponding stream still in the system. The value of the deadline credit counter is limited by $D\_CR^j_{max}$, for some $D\_CR^j_{max} > T^j - D^j$ for stream j, setting a limit on the amount of positive deadline credit build up or number of ADUs that can be sent ahead of schedule. $D\_CR^j_{max}$ represents the capacity of the corresponding receiving buffer. It is assumed that the receiving buffer of stream j can store a maximum of n data units. This corresponds to a session duration of nT^j slots. So the deadline credit counter is upper bounded by $D\_CR^j \leq (n+1)T^j = 3DD\_CR^j_{max}$ slots; $D\_CR^j_{max}$ is known *a priori*. The additional unit is added to account for the ADU of the current periodic interval.

At the beginning of session j, D_CR^j is set equal to zero. During the session D_CR^j is updated as follows:

(a) During every elapsed slot: $D\_CR^j = D\_CR^j - 1$. That is, the ADT is reduced by 1 for every slot that passes by.

(b) At the end of an examination slot for stream j: $D\_CR^j = D\_CR^j + (k+m)T^j$, where k is all the dropped ADUs (which are all expired) and the binary (0/1) variant m is 1 only if : (i) $D\_CR^j + (k+m)T^j \leq D\_CR^j_{max}$ and (ii) $D\_CR^j + (k+m)T^j + D^j \geq \hat{P}^j_i$, where P^j_i is the non expired ADU at the head of stream j. Step (b) "adjusts" the context of D_CR^j to describe the ADT of the ADU at the head of stream j after removing the dropped/scheduled for service ADUs of stream j. Step (b-(i)) checks the buffer availability and step (b-(ii)) checks the deadline condition for the ADU at the head of stream j.

The second counter, L_CR^j, is set up to count the number of ADUs that are dropped from stream j up until the last examination slot for stream j. L_CR^j is updated at examination slots for stream j only as follows: $L\_CR^j = L\_CR^j + k$, where k is equal to the number of ADUs of stream j dropped in the current examination slot for stream j.

In order to guarantee that the available bandwidth will be distributed fairly between all competing streams, the two metrics are combined into a single metric, called priority index, that aims to capture both the current deadline credit of the stream and its cumulative losses. The priority index counter P_CR^j is defined

$$P\_CR^j = \frac{D\_CR^j - T^j \cdot L\_CR^j}{T^j}$$

.

The priority index counter represents the relative priority of competing data streams. The smaller the value of the priority index counter of a stream the higher the priority of the stream with respect to the others. This will be the stream with the most losses, if any, and among streams with no losses or the same losses it will be the stream with the lowest deadline credit.

The flowchart in Fig. 1 summarises the DC algorithm. A more detailed and insightful description of the DC policy may be found in [7].

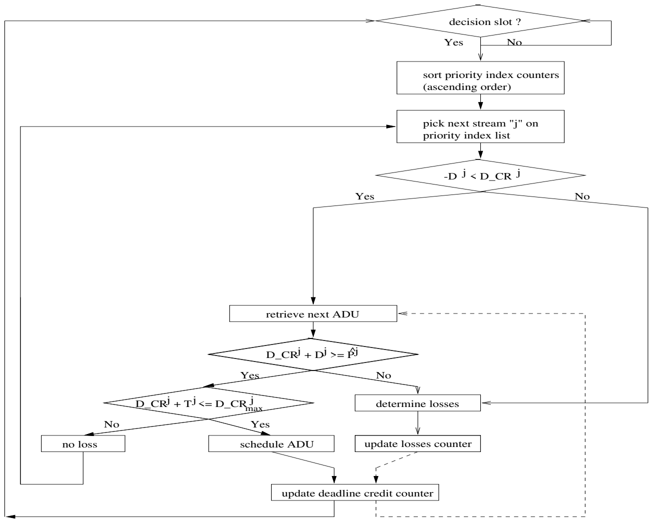

Figure 1: The DC scheduling algorithm flowchart.

3 Numerical results and discussion

The performance of the DC policy is investigated in terms of its fairness and induced losses. It is compared to the FIFO policy, which is of much lower complexity, and an approximation (described later in the section) of the CRTT policy as described in [3] and [6]. The performance is measured in terms of the ratio of the induced ADU losses to the total number of ADUs offered to the scheduler versus increasing system utilisation. The average bandwidth required by all streams, U_{req}, is defined as the sum of the average cell rate generation of all streams. So for a total of N data streams:

$$U_{req} = \sum_{j=0}^{N-1} \frac{\hat{P}^j_{ave}}{T^j}$$

Due to space limitation a selective list of results is presented.

Fig. 2 illustrates the fairness of the DC policy by showing the deadline credit counter for $S = 4$ streams with $T = 40$ slots, $D\_CR_{max} = 500$ slots and $D = T$. In this case, U_{req} is at 85% so 15% of the total capacity is used as *credit build up capacity*.

The performance of the DC policy is compared against that of the FIFO policy which is of much lower complexity and does not exploit the available bandwidth when the system is underutilised. For the FIFO implementation it is assumed that ADUs arrive at the scheduler in a random order at a rate of one ADU per stream and period. The p-FIFO version serves one ADU per period from each stream on a FIFO basis without considering the deadlines of the ADUs. The d-FIFO version discards the ADUs which can not meet their deadline. Fig. 3 shows the losses induced by p-FIFO, d-FIFO and DC for $S = 4$ data streams with $D\_CR_{max} = 100$ and 500 with the period varying in the range of $T = 25$ to $T = 100$ slots. The losses reach one for the p-FIFO implementation when the utilisation reaches 100%. The losses for the d-FIFO follow the p-FIFO curve for a system utilisation between 50% and 70% and then grow at a lower rate than the p-FIFO and converge to the DC curve for utilisations higher than 130%. The losses under the DC policy are at least 4 orders of magnitude less than the losses induced by the FIFO approaches for utilisations less than 90%.

In the last figure the DC policy is compared to an approximation of the CRTT scheme proposed in [3] and [6]. The CRTT scheme is simulated here in the following way: A number of ADUs from each stream are initially transmitted in advance (prior to the commencement of playback) which corresponds to the initial $BUILD\_UP$. The remaining ADUs are transmitted at a pre-defined constant rate of r cells/period during playback. The rate r is guaranteed for each stream. A stream can not transmit more than r cells/period. Fig. 4 shows plots for three different values of $BUILD\_UP = 100, 500, 1000$ slots which correspond to 1, 5 and 10 ADUs respectively. In the plots the rate r takes values in the range of $r = 1.0$(average ADU length/period) to $r = 1.3$(average ADU length/period) for each stream. For each rate, the losses are plotted as a function of the increasing number of multiplexed streams. For a given rate r, the losses fluctuate around an average value as the number of streams increase which is expected as each competing stream has a guaranteed amount of bandwidth (r). Since a rate is guaranteed for each stream, as r increases the total number of streams that can be served by the system decreases. Notice that the loss curves extend up to the maximum number of streams that can be accommodated by the system for a given r and that as r increases the loss curves become shorter in length. The losses for rate $r = 1.0$(average ADU length/period) are higher than 90% and the maximum number of streams than can be served is 10. As r takes higher values the average value of the losses decreases and the maximum number of streams that can be served ranges between 7 and 9.

For the case of $BUILD\_UP = 100$, which corresponds to an initial build up of 1 ADU, the DC policy is implemented for $D\_CR_{max} = 100$ and 200 slots, which corresponds to a buffer capacity of 1 ADU (limiting case) and 2 ADUs respectively. For $D\_CR_{max} = 100$ the performance of the DC policy is comparable to that of the CRTT implementation for various values of r. For the case of $D\_CR_{max} = 200$, however, the DC policy provides the same or lower levels of losses for the same number of multiplexed streams.

For the case of $BUILD\_UP = 500$, and 1000 which corresponds to an initial build up 5 and 10 ADUs respectively, the DC policy is implemented for $D\_CR_{max} = 500$ slots, which corresponds to a buffer capacity of 5 ADUs. In both plots, when compared against the CRTT implementation for a rate $r = 1.1$(average ADU length/period) the DC policy has the same level of losses for $S = 9$ streams and lower levels of losses by at least two orders of magnitude for a smaller number of streams.

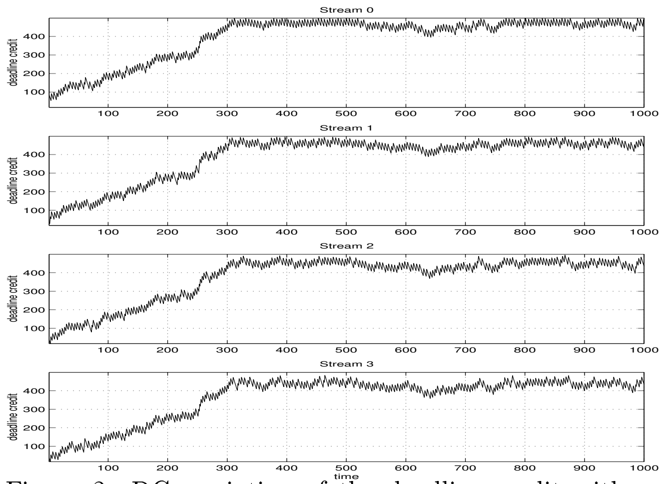

Figure 2: DC: variation of the deadline credit with time, $U_{req} = 85\%$.

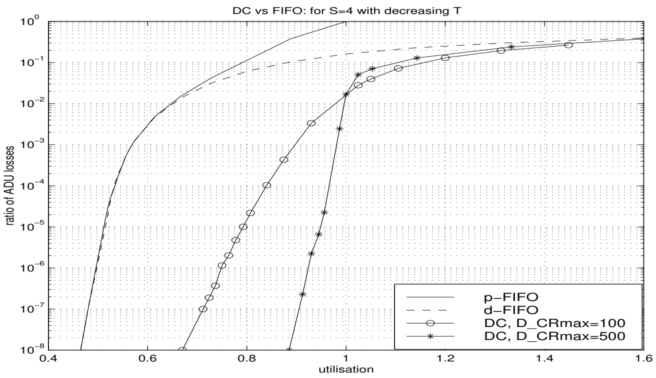

Figure 3: DC vs FIFO: ADU losses vs system utilisation, $T = 25 - 100$ slots.

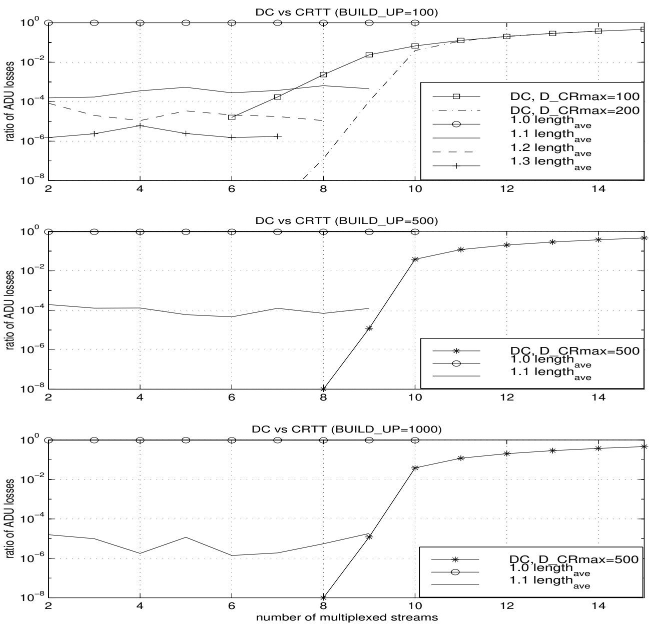

Figure 4: DC vs CRTT: ADU losses vs number of multiplexed streams.

References

[1] Klara Nahrstedt Ian R. Philp and Jane W.S. Liu. Scheduling and Buffer Management for Soft-Real-Time VBR Traffic in Packet-Switched Networks. *Technical Report, University of Illinois at Urbana-Champaign, Ill*, 1996.

[2] George Kesidis. *ATM Network Performance.* Kluwer Academic Publisher, 1996.

[3] J.M. Mc Manus and K.W. Ross. Video on Demand over ATM: Constant-Rate Transmission and Transport. *Technical Report, University of Pennsylvania, Philadelphia, PA, http://www.seas.upenn.edu/∼ross/*, November 1995.

[4] Zoe Antoniou and Ioannis Stavrakakis. Earliest Due First Scheduling for Application-Level QoS Delivery. *Proceedings of IEEE Conference on Protocols for Multimedia Systems - Multimedia Networking, pp.172-182, Santiago, Chile*, November 1997.

[5] Sujata Banerjee. Translating Application Requirements to ATM Cell Level Requirements. *Proceesings of IEEE International Conference on Communications*, 1997.

[6] J.M. Mc Manus and K.W. Ross. Prerecorded VBR Sources in ATM Networks: Piecewise-Constant-Rate Transmission and Transport. *Technical Report, University of Pennsylvania, Philadelphia, PA, http://www.seas.upenn.edu/∼ross/*, September 1995.

[7] Z. Antoniou and I. Stavrakakis. Deadline Credit Scheduling Policy for Prerecorded Sources. *Report, CDSP Centre, Northeastern University*, 1999.