ISSN:0975-9646

Pragati Gautam et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 1914-1917

Preventing Packet Loss in Aggregated Streaming Data

Pragati Gautam [1],C. Santhanakrishnan[1]

[1]Post Graduate Student ,[2]Assistant Professor
Computer Science and Engineering
SRM University
Kattankulathur, Chennai, Tamil Nadu, India

***Abstract*-Introducing the problem of managing a buffer of bounded space, where arriving packets have dependencies among them. Multimedia applications require transmission of streaming video from a server to a client across an internetwork. In many cases loss may be unavoidable due to congestion or heterogeneous nature of the network. In this paper considered online policies for selective frame discard and analyse their performance by means of competitive analysis. Our online algorithm used to prove the upper bound is non pre-emptive no packet admitted to the buffer. Algorithm is, in case of an overflow, to prefer keeping packets from frames for which it has already delivered many packets, and dropping packets from frames with fewer packets already delivered. This algorithm essentially tries on effort already invested in delivering earlier packets of a frame.**

***Index Terms*—** buffer, streaming styling, competitive analysis, quality of service, variable bit rate.

I. INTRODUCTION

In networked applications, application layer data frames are split into smaller sized packets, when sent across the network. The receiver side can make use of the data only if it receives *all* the packets of a frame. Higher-level mechanisms in the protocol stack usually handle retransmissions of lost packets, in order to provide adequate performance for the application. The exact dependency structure of the data stream depends when data is encoded forms. In MPEG video encoding schemes, where successfully decoding a frame might depend on successfully decoding other previous/later frames). in MPEG video encoding schemes, where successfully decoding a frame Might depend on successfully decoding other previous/later frames). When we consider the real time traffic. The common approach, to deal with packet losses is to employ proactive encoding schemes; however, this approach has its limitations in several networking environments. Specifically, some environments (e.g., wireless networks). Also, in some scenarios where traffic may traverse bottleneck links, the effect of coding Diminishes substantially, since the bottleneck fully Determines the loss characteristics incurred by the traffic. In this work we focus on FIFO buffer architecture, the features of (a) it is simple, (b) it maintains the arrival order of incoming traffic, hence avoiding the need for mechanisms that deal with packet reordering, and (c) it provides simple and reliable delay bounds. main causes for packet loss in networks are buffer Overflows due to congestion. in such case the underlying traffic has inter-packet dependencies, indiscriminately dropping packets upon overflow may result poor performance. Actually our aim is packets must be delivered in proper manner and the needs will get effective good put.i.e. the amount of data can be decoded effectively at the receiving end .among all packets if one packet is dropped the result is zero good put method to decide which packets to drop in case of overflow is critically important to the performance of the system, bearing in mind that such a decision might effect other packets which have already been forwarded, or packets that have not yet arrived.eg [1] goal into devise methods that maximize the good put of successfully delivered traffic, captured by the number of useful frames delivered. We consider the problem of buffer management of multiple data streams in scenarios where traffic has inter packet dependencies. Some guidelines provide when we design an algorithm, that algorithm provide high performance in the terms of good put. The exact dependency structure of the data stream depends on the encoding used and it may consist of a dependency structure, where frames are independent of each other, and the only dependencies are among packets corresponding to the same frame. Concentrate on three different queuing policies. The nonpreemptive policy transmits all packets admitted into the queue; observe, under this policy, the queue Can easily maintain a FIFO order. The FIFO preemptive policy is allowed to drop packets already admitted to the queue. The bounded delay model, on the other hand, transmits packets in any order, but each packet must be transmitted before a fixed deadline or else the packet is lost.

Generally, content and service providers are averse to provide Multimedia data, such as pre-encoded video for streaming Services, separately for wired and wireless clients. It's expected that a huge amount of content is exclusively stored on a server in the "wired" Internet—accessed by both fixed and wireless clients. Packets delays and loss are very common in nowadays IP-based packet networks. However, due to predictive video coding, lost IP packets result not only in decoding errors of the current frame, but also in quality degradation of subsequent frames included in the dependency chain. Whereas packet Losses and delays in fixed Internet mainly result from network congestion, wireless transmission Packet losses and delays usually. Resulting from multipath propagation, scattering, and fading, and to guarantee an error-free Reception of IP-packets at the expense of delay. As a simple mean decoder buffer in combination with an initial playback delay to smooth the bit rate variations caused by the transmission channel [1]. In this work, we will concentrate on the transmission of variable bit rate (VBR) encoded video over VBR channels. After formulated the exact problem and discussing related work, we will show that the separation between delay jitter buffer and decoder buffer is, in general, suboptimal for VBR video transmitted Over VBR channels. Will define the minimum initial delay and the minimum required buffer for a given video sequence and a deterministic VBR channel. In addition, provide some probabilistic statements in the case that we have a random behavior of the channel bit rate. A specific example tailored to wireless video streaming will be discussed in greater detail and bounds will be derived which allow to guarantee a certain quality-of-service., even for random Variable bit rate channels ina wireless environment.

Our results show that the algorithmic solutions in maximizing Weighted throughput as well as their computational Complexities are significantly different from those optimizing throughput of uniform-value packets.

Competitive Analysis: We measure the performance of our algorithms using competitive analysis [1, 2]. In competitive analysis the performance of an online policy is compared with that of an optimal offline policy, this knows in advance the entire sequence of frame arrivals Competitive analysis is a approach for Internet traffic, which is not predictable. The advantage of competitive analysis is that a uniform Performance is provided for all input sequences.

2.0 Our Contribution

In this paper, introducing problem of online set packing, present Randomized distributed algorithms for it, and prove upper bounds on the competitive ratio Of any online algorithm for it .two guidelines of this algorithm

1) Once a frame has a packet admitted toThe buffer, make sure that every attempt possible to deliver the complete deliver the whole frame as soon as possible.
2) We used to give priority preference of packets.
3) We analyze the performance of our algorithm, and show That for *any* traffic the ratio between its performance and That of an optimal algorithm is always bounded.
4) We prove upper bounds on the performance of any buffer Management algorithm.

2.1 *Previous Work*

The problem of packet forwarding with inter-packet dependencies is set in the context of video traffic. For this recently no. of procedure done for packet discarding schemes proposed [3] .In these cases too many packets are dropped some times entire frame dropped. then they evaluated Markovian video sources.addional works that consider the competitive algorithms.ie.provide the quality of service been studied here [3] no of them addresses interpacket dependencies.[4] The work done in proactive schemes Although model assumes no redundancy we believe a better understanding of this basic scenario is the starting point for designing algorithms that additionally account for proactive coding.

Many of these works study systems that should provide some Quality-of-Service guaranteed to the underlying traffic to the buffer capacity and FIFO order constraints the FIFO model, it is typically assumed that each packet has a value, and the goal of the algorithm is to maximize the total value of delivered packets [4, 5]. In case a single buffer best known competitive ratio for algorithms model provides an abstraction of the buffer overflow management problem in such systems and our algorithms can be seen as treating such settings from a more systems oriented viewpoint.

3.0 Model

System consists N streams of unit size packets, denoted by $s_1 \ldots\ldots s_n$. Every stream s_n is viewed as a sequence of frames, F_i^n and consisting of a sequence of Exactly k packets, $P_1^{n,i}, \ldots\ldots P_k^{n,i}$. A packet $P_j^{n,i}$ is referred as jth packet of frame and its arrival time is denoted by a($P_j^{n,i}$). When we refer into the packets sometimes omit the frame index and used the notation $\{p_j^n\}$ j= 1, 2… when refer to the sequence of corresponding stream s_n.whereas p_j^n denotes the jth stream s_n and jmod k packet of frame $f_{\lceil j/k \rceil}^n$ .packets of arrived in that order .i.e. a(p_j^n) $\leq a(\, p_{j+1}^n)$ for all j. Notation implies for the following arrival of sequence of the stream s_n consist r_n $packets$ $p_0^n \ldots.. p_{k-1}^n, \ldots, p_{r_{n-1}k}^n \cdots p_{r_n k-1}^n$. arrival of packets from different streams implies a finite arrival sequence σ of the aggregated streams, which is the Interleaving of the arrival sequences of the individual streams.

3.1 *Buffer Model*

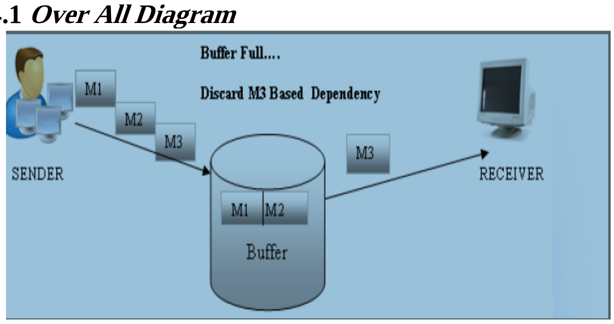
Packets arrive at a FIFO buffer can store **B**$\geq K$ packets transmit one packet per cycle Initially the buffer is empty, their consisting two steps first step delivery step is the arrival step and In the second sub step, called the arrival step. At the discretion of the buffer Management algorithm, some packets may be dropped, while other packets are stored in the buffer. We refer to a Time interval (a,…,b) as the sequence of cycles a + 1, . . . ,b

4.0 System Technique

In this system, we consider the problem of buffer management of multiple data streams in scenarios where traffic has inter-packet dependencies. We provide guidelines for designing algorithms that are guaranteed to provide high performance in terms of good put. Our approach and analysis provide bounds on the performance of the proposed algorithms for any traffic.

arrival pattern, without requiring any deterministic assumptions on the processes generating the traffic. In effect, we commonly consider the traffic to be adversarial. Different from works which focused on deterministic traffic models, our approach is orthogonal to works that aim at exploiting try to analyze the tradeoff between available network resources (e.g., in terms of the buffer size available) and system performance.

4.1 *Over All Diagram*



4.1 *System Technique Explanation*

Online Algorithm

Algorithm is one that can process its input piece-by-piece in a serial fashion, the order that the input is fed to the algorithm, without knowing the entire input available from the start. In contrast, an **offline algorithm** is given the whole problem data from the beginning .Because it doesn't know the whole input, an online algorithm is forced to make decisions that may later turn out not to be optimal, and the study of online algorithms has focused on the quality of decision-making that is possible in this setting. Competitive analysis formalizes idea by comparing the relative performance of an online and offline algorithm for the same instance problem. Specifically, the competitive ratio of an algorithm, is defined as the worst-case ratio of its cost divided by the optimal cost, over all possible inputs. The competitive ratio of an online problem is the best competitive ratio achieved by an online algorithm. Intuitively, the competitive ratio of an algorithm gives a measure on the quality of solutions produced by this algorithm, while the competitive ratio of a problem showed to the importance of knowing the future for this problem. For other points of view on online inputs to algorithms, see streaming algorithm and whereas , dynamic algorithm and online algorithm. An alternative analysis of the problem can be made with the help of competitive analysis. For this method of analysis, the offline algorithm knows in advance which edges will fail and the goal is to minimize the ratio between the online and offline algorithms' performance.

Scheduling packets with deadlines is essentially an Online decision problem. Evaluate the worst-case Performance of an online algorithm lacking of future Input information, we compare it with an optimal offline algorithm. The offline algorithm is a clairvoyant algorithm, empowered to know the whole input sequence (including the fading states of the channel, the packet sequence, and all packets' characteristics) in advance to make its decision. In contrast to stochastic algorithms that provide statistical guarantees under some mild assumptions on input Sequences, competitive online algorithms guarantee the worst-case performance.

The upper bounds of competitive ratios are achieved by some known online algorithms. A competitive ratio less than the lower bound is not reachable by any online algorithm. An online algorithm is said to be optimal if its competitive ratio reaches the lower bound. If the additive constant δ is no larger than 0, the online algorithm ON is called strictly k-competitive. Note that a randomized Algorithm does not depend on any assumptions on the input Sequence and the randomness r is internal to the algorithm.

Competitiveness has been widely accepted as the metric to measure an online algorithm's worst-case performance in theoretical computer science and operations research [4]. In this section, we design and analyze some competitive online scheduling algorithms for maximizing weighted throughput

Our online algorithm used to prove the upper bound is non preemptive: no packet admitted to the buffer. Algorithm is, in case of an overflow, to prefer keeping packets from frames for which it has already delivered many packets, and dropping packets from frames with fewer packets already delivered. This algorithm essentially tries on effort already invested in delivering earlier packets of a frame.

Consider 2 packets transmission,

Step 1: First, 2-packets whose 1-packets were delivered are taken.

Step 2: If there is additional room, complete frames (both their 1-packet and 2-packet) are taken.

Step 3: Finally, remaining 1-packets fill the leftover space if any.

5.0 Online Computation, Competitive Analysis, And Priority Algorithms

This section introduced online algorithms and competitive analysis. The framework For deriving lower bounds for priority algorithms is borrowed from competitive analysis of Online algorithms. here two examples of deriving lower bounds for deterministic And randomized paging algorithms and then give an example of deriving a lower bound on the approximation ratio achieved by fixed priority algorithm

Online computation, an algorithm must produce a sequence of decisions that will have Be made based on past events without secure information about the future. Such an Algorithm is called an online algorithm. Priority algorithms resemble online algorithms. Both algorithms do not see the whole instance; rather they observe the input one item at a time. Both algorithms must make an irrevocable decision about a data item, based on the partial input seen so far. The differences between online algorithms and priority Algorithms are in the order in which the algorithms see the input. Priority algorithms can use arbitrarily complex functions to order the data items. In the case of online Algorithms, the Adversary or other constraints the order.

5.1 *Competitive Ratio*

a) Competitive Ratio Of Weightpriority Is $O((kMB + M)^{k+1}$

Given high description of analysis. we like delivered frame by an optimal solution to frames delivered by wp.[8] To this end considering partition of time intervals, and identify every interval with the highest ranking over the intervals map each interval to strictly higher ranking interval in which frame successfully delivered ,Cumulating interval in which frames successfully delivered [7]. The number of frames successfully delivered by an optimal solution. the number of intervals mapped to any Single interval, one obtains a k-height K-ary tree-like Structure underlying the mappings of intervals, which Implies the required result.

b) Upper Bound

Analysis of Round Robin Policy that equally divides the buffer into n partitions of size B/M Where M is the number of different packet values. Every partition assign a different value, and only packets value is accepted into this partition, in greedy manner. The partitions take turns in sending packets. If a partition's turn to send a packet arrives, but it is empty, its passes to the next partition. We simulated the Round Robin method using a single queue as followed. Virtually keep track of the current state of the Round Robin method, used virtual state to decide whether or not to accept the currently incoming packets. Since the Round Robin Method transmits 1 packet per time step, In the real queue size coincides that the virtual queue Size, and queues accept the same set of packets.

6.0 Scope Of The Project

Our scope is transfer packets are very effectively and also receive original data clearly. This system split data as multiple packets and transmitted to buffer, if buffer having sufficient space to allow to store and delivery to receiver. Suppose buffer having insufficient space, it is not allow to storing any packets. This way to improve the performance of sending and receiving packets are correct.

Flow:

Step 1: Split sending data into no of packets.
Step 2: Calculate the size of packets and count.
Step 3: Check buffer size.
Step 4: Transfer packets to receiver

7.0 Conclusion

In our System provided guidelines for the design of algorithms and analyzed the performance of buffer management algorithm, both from a worst case competitive approach. We provided guarantees on its performance under any traffic conditions by proving it has a bounded competitive ratio. We also showed that the competitive ratio of any algorithm for our problem might degrade linearly as a function of the number of streams in the traffic.

References

[1] V. Varsa and I. Curcio, “Transparent end-to-end packet switched streaming service (PSS); RTP usage model (Release 5),” 3GPP TR26.937 V1.4.0, 2003.

[2] A. Borodin and R. El-Yaniv. Online Computation and Competitive Analysis. Cambridge University Press, 1998.

[3] D. Sleator and R. Tarjan. Amortized Efficiency of List Update and Paging Rules. CACM 28, pp. 202-208, 1985

[4]S. Ramanathan, P. . . Rangan, and H. M. VIN. Frame-Induced Packet Discarding. An Efficient Qos Management Strategy for Video Networking. Fourth International Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster University, November 1993.

[5] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko, “Buffer overflow management in QoS switches,” *SIAM Journal on Computing*, vol. 33, no. 3, pp. 563–583, 2004.

[6] Y. Mansour, B. Patt-Shamir, and O. Lapid. Optimal smoothing schedules for real-time streams. Distributed Computing, 17(1):77–89, 2004.

[7] Priority Algorithms Sashka Davis University of California, San Diego sdavis@cs.ucsd.edu June 2, 2003.

[8] Gabriel Scalosub, Peter Marbach, and Jo¨ rg Liebeherr, Fellow, IEEE . more information any other computing topic, please visit Digital Library at www.computer.org/publications/dlib.

[9] en.wikipedia.org/wiki/Online algorithm El-Yaniv, R. (1998). Online Computation and Competitive Analysis. Cambridge University Press. ISBN 0-521-56392-5

[10] *Bernd Girod, Mark Kalman, Yi J. Liang, Rui Zhang* Information Systems Laboratory Department of Electrical Engineering, Stanford University Stanford, CA 94305

[11] Y. Wang, and Q. Zhu, “Error control and concealment for video communication: a review,” *Proceedings of the IEEE*,vol. 86:5, p. 974-97, May 1998.