Master's Thesis

A Distributed Source Control for Multimedia ATM Networks

by M.U. Lynch
Advisor: E. Geraniotis

CSHCN M.S. 93-2
(ISR M.S. 93-2)

CENTER FOR SATELLITE AND HYBRID COMMUNICATION NETWORKS

The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.

Web site http://www.isr.umd.edu/CSHCN/

Abstract

Title of Thesis: A Distributed Source Control for Multimedia ATM Networks

Name of Degree Candidate: Michael Eugene Lynch

Degree and Year: Master of Science, 1992

Thesis directed by: Evaggelos Geraniotis, Professor, Electrical Engineering

The effects of multimedia data such as voice, audio, video, still-image, text, and graphics on high speed networks need to be identified for the purpose of developing efficient multimedia network designs and implementations. In this thesis, we identify a model for the most demanding multimedia type: motion video. Then, we define a network protocol for handling the bursty nature of the multimedia type. The protocol is placed at the source and is defined as an adaptive distributed source control (DSC) to adjust to the particulars of the given video call's virtual channel. The internal network protocol is also defined to work closely with the source nodes. In a computer simulation, the performance of adaptive DSC protocol in a multimedia ATM network is contrasted with non-adaptive DSC protocol performance. By adding bursty video data types, a decrease in the network protocol performance is shown while lowering DSC settings improves performance. On the basis of these results, an overall adaptive DSC protocol strategy is shown to be the best approach in implementing the DSC protocol, especially when the statistics of the source are not known prior to transmission.

A DISTRIBUTED SOURCE CONTROL FOR MULTIMEDIA ATM NETWORKS

Michael E. Lynch

A Distributed Source Control for Multimedia ATM Networks

By

Michael Eugene Lynch

Thesis submitted to the Faculty of the Graduate School
of The University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1992

Advisory Committee:

Professor Evaggelos Geraniotis, Chairman/Advisor
Professor Anthony Ephremides
Associate Professor Ahmet Oruc

Table of Contents

List of Figures

List of Tables

CHAPTER 1

BACKGROUND AND INTRODUCTION

1.1 Background

Networks are becoming faster and they continue to carry an ever increasing set of diverse data types with wide ranges of bandwidth and Quality of Service (QOS) requirements. The trend is for faster network speeds (1Gbps & higher). This increase in network speed allows for increased bandwidth utilization based on the statistical multiplexing of the various types of multimedia data, while potentially delivering a relatively high number of digital video and audio sources. Also, the increased network speeds allow the option of transmitting data such as motion video with various quality attributes such as HDTV (very high quality), Commercial broadcast (high quality), Business broadcast (medium quality), MPEG (VCR quality), and video telephony (low quality). With higher speeds allowing for additional capabilities, the management of the network becomes more complex with more bandwidth to manage. The increase in network bandwidth allows for more combinations of various types of multimedia data being transmitted at any one time. New network management schemes are required for the technical challenges presented by the faster network speeds and additional data types.

Asynchronous Transfer Mode (ATM) is an emerging switching and transport protocol deemed to be a prime candidate for Broadband Integrated Services Digital

Network (B-ISDN). The thesis has focused on the basic concepts of ATM transport mechanism for network simulation. The ATM protocol was designed with the consideration of servicing various data types from the onset. The ATM packet (or cell) is short in length and fixed at 53 (48 bytes payload and 5 bytes header). The fixed-length cells simplify the hardware design for a high speed ATM switch. In addition, the short cell size reduces the delay and the delay variance (jitter) for delay sensitive source types such as video and audio, which are considered within this thesis. The ATM protocol is considered to be at level 2 in the OSI networking model while the Synchronous Optical NETwork (Synchronous Digital Hierarchy) (SONET/SDH) transmission standard is at level 1 (physical layer).

Bursty data types such as data files and voice sources with talkspurt and silence detection have received a significant amount of focus by the network research community over the years in terms of how these sources affect the network performance. Lately, due to technology improvements in network speeds and video encoding algorithms along with the momentum by users to switch to land-based networks for transmitting motion video (i.e., video teleconferencing applications), the research community is quickly focusing more attention on network performance due to bursty video sources. This thesis is addressing network performance based on multimedia input types including data, audio, and video. Though each data type is represented within the network simulation, the focus is on the video sources due to the relatively limited knowledge base for real-time video applications in an ATM network.

1.2 Introduction

The prelude to faster networks provides new technical challenges for networking that were either non-existent or of a lower priority concern with slower speed networks. At the higher speeds, the network propagation delays for wide area networks (WANs) start dominating the queuing delays. This factor plays a role in the decreased performance of reactive network management control schemes for high speed networks. Network management control schemes that emphasize congestion reaction versus congestion prevention have an inherent latency that may allow for a large number of cells to be discarded before the required corrective action is in effect. Also, due to the speed of the network, the congestion may be cleared by itself prior to the arrival of any corrective action. For higher speed networks, congestion avoidance may be the preferred method for implementation instead of reactive control that occurs after the detection of congestion. Existing network management schemes have also emphasized the discarding of cells or lowering the QOS to maintain the negotiated network transmission rate. This thesis proposes an alternative to this type of network management control called the Distributed Source Control (DSC). The DSC protocol emphasizes the prevention of cell discarding and the maintenance of the QOS by a traffic shaping technique that smoothes the burstiness of various data types.

Network management control schemes have been challenged in the past with bursty data type sources. However, the advent of bursty video sources with variable bit rate (VBR) coding, real-time requirements (new and variable batch size of cells every video frame; NTSC frame rate 1/30 sec), and a relatively large bandwidth requirement (kilobits for low-end video teleconferencing to megabits for high end video broadcasts), has pushed many existing network management control algorithms past their intended design limit. Any new network management scheme that handles bursty video sources will be prime candidates to handle other types of bursty data. We will be working directly with a couple of video source models to test the performance of network management schemes defined within this thesis.

1.2.1 Overview of Network Management Scheme Options from Other Related Work

To the network designer, several existing network management schemes from other research work are available for policing the network sources:

- Leaky Bucket [1], [2], [3]
- Rate Control [4]
- Jumping Window [3]
- Triggered Jumping Window [3]
- Rectangular Sliding Window [2], [3]
- Triangular Sliding Window [2], [3], and
- Exponentially Weighted Moving Average [2], [3]
- Traffic Smoothing [5]

Other than rate control and traffic smoothing, all of these policing schemes emphasize the dropping of cells if certain thresholds are exceeded by the source. Rate control emphasizes reducing the quality of the source by lowering the source bit rate if the network encounters congestion. All of these policing schemes (leaky bucket, window control, and rate control) have a role in network management. However, in contrast to the other policing schemes that emphasize dropping of cells or reducing the source rate, which affects the quality of video and audio sources, this thesis proposes an alternate approach prior to these schemes becoming active.

The proposed approach is a hybrid "policing and rate control" scheme that emphasizes traffic shaping and de-emphasizes dropping of cells. The hybrid approach called Distributed Source Control (DSC) attempts to shape the distribution of cells entering the network from variable bit rate sources by smoothing the burstiness of the source. The traffic smoothing represented by the non-adaptive DSC approach was originally presented in [5]. The non-adaptive DSC approach is enhanced by defining an adaptive DSC approach that adjusts to the particulars of source call statistics.

1.2.1.1 Overview of the Non-Adaptive DSC Network Management Scheme

The non-adaptive DSC protocol developed in [5] was defined for a high-speed, non real-time data source type. This data source type was identified as a source with large file transfers, image retrievals, etc. The non-adaptive DSC involved the negotiation of two parameters:

- DSC Window (cells)
- Smoothing Interval

The DSC protocol was defined to be network frequency independent by the incorporation of a frequency-memory constraint during the negotiation of the bandwidth management parameters. The frequency-memory constraint is a function that relates the smoothing interval to the operating frequency of the network and the amount of buffer memory that can be made available at that network frequency. The DSC forms a ratio of the window cells to smoothing interval that can be made constant and represents a negotiated transmission rate. However, the non-adaptive DSC protocol did not directly address requirements of bursty, real-time data sources and did not address how the initial transmission rate was negotiated when the source statistics are not known prior to transmission.

1.2.2 Thesis Goals

This thesis has the following major goals:

- Obtain network simulation results based on an ATM network with the inclusion of all three major multimedia types: data, video, and audio
- Identify a robust distributed network management scheme including both network and network edge management that includes the advantages of both schemes
- Enhance a non-adaptive DSC protocol by defining an adaptive DSC protocol that can handle real-time data with an emphasis on controlling bursty video source
- Compare the performance of a non-adaptive and adaptive DSC algorithms

Multimedia communication is addressed in general within this thesis. However, the focus is on the video sources due to the relatively limited knowledge base at a detailed level for real-time video applications in an ATM network. Bursty video sources with variable bit rate (VBR) coding, real-time requirements, and a relatively large bandwidth requirement, present significant challenges to network management control schemes. Several video source models will be considered for testing the performance of network management schemes proposed within this thesis.

The thesis will take a system approach to the network management of multimedia data. As a result, we propose a combined approach of Network Edge and Network Management for multimedia networks. The Network Edge portion will help to reduce the occurrences of network congestion caused by sources, while the Network Management portion will help to minimize the effects of network congestion once they do occur in the network (Note: Network Management in this thesis refers to the management of the network once data has entered the internal section of the network). The proposed approach within this thesis involves a combination of distributed network management control algorithms at the network edge and within the network provides the most robust approach in controlling the multimedia network traffic.

This thesis intends to build upon some of the basic concepts presented in [5] for controlling non real-time bursty data sources. Also, in contrast to the other policing schemes that emphasize dropping of cells or reducing the source rate, this thesis proposes an alternate approach that emphasizes traffic shaping and de-emphasizes dropping of cells. This approach attempts to shape the distribution of cells entering the network from real-time, variable bit rate sources by smoothing the burstiness of the source. The non-adaptive DSC approach is enhanced by defining an adaptive DSC approach that adjusts to the particulars of source call statistics.

If this approach is implemented today, the cost of such an implementation will be relatively high when compared to some other approaches, which simplify both the computation complexity and the amount of hardware such as memory to implement any given network management control algorithm. This thesis attempts to focus on the research of a baseline network management control and protocol for multimedia network communication. The intent is to lay a foundation that others may be able to refine or redefine to improve multimedia networks. Technology innovations, as the years go by, continue to drive costs of memory down and continue to increase the throughput and computation power of central processing units for all processing platforms. Since this thesis involves research that represents a possible future implementation, the premise here is that when this multimedia network approach or a similar approach is feasible for commercial implementation, the technology will be such that some or all of the robust network approaches proposed with this thesis will be economical to implement.

1.2.3 Thesis Organization

This thesis is organized as follows: the video model options considered for inclusion in the ATM network simulation are analyzed in Chapter 2. The non-adaptive DSC protocol from [5] is expanded by the adaptive DSC protocol that is defined in Chapter 3. In Chapter 4, the performance of the non-adaptive DSC protocol with bursty video sources is compared to the performance that includes a fixed bit rate video source. This chapter also includes the performance comparison between the non-adaptive and the adaptive DSC algorithms. The overall conclusions of this thesis are presented in Chapter 5. Finally, an overview of the simulation setup is included in Appendix A.

CHAPTER 2

VIDEO SOURCE MODELING

With the improvements in network speeds and the increased emphasis on multimedia communications, researchers have identified and defined various source model types. Data Source Modeling has been researched extensively over the last two decades. Audio Source Modeling has made significant progress during the last decade. In contrast to these types of data types, video source modeling is still being researched and improved. The options for video source modeling are limited in terms of number of source model candidates and the type of video source being modeled. Currently, the video source models concentrate on video teleconferencing examples. These video teleconferencing examples are limited in the types of motion that occur during the video call. The video examples limit the activity to a "talking head", which is typical in many video teleconferencing applications. However, the video source models do not include any camera zoom and panning effects that usually occur in a video teleconference call that has two or more people at the transmit site. The options for video source modeling considered in this thesis are discussed below.

2.1 Fixed Bit Rate (FBR) Video Modeling

FBR Video Source Modeling produces a constant bit rate for encoding video information:

$$\text{NTSC frame rate} = 1/30 \text{ second/frame} \tag{2.1}$$

$$\text{Coding rate} = C_r \text{ bits per frame} \tag{2.2}$$

$$\text{FBR} = C_r \times 30 \text{ bits/sec} \tag{2.3}$$

For encoding video information, there are fewer available bits per video frame with a fixed, low video bit rate encoder than a fixed, high video bit rate encoder. With a fixed rate, the distortion will vary with each video frame depending upon the amount of changes within the video frame. At a low bit rate, the distortion will vary significantly on the rate-distortion curve over a range of distortion that is relatively large on the rate-distortion curve. Likewise, at a fixed, high video bit rate, the distortion will also vary significantly on the rate-distortion curve. However, the range of distortion will be at the low-end of distortion spectrum. Figure 2.1 illustrates the above discussion between fixed, low and high bit rate encoders. The Distributed Source Control protocol performance has one simulation run with a configuration that uses a single video source per 150Mbps link. The video source produces 45Mbps.

The packetization interval (P_r) for a FBR video source on an ATM network is defined as follows:

$$V_r = \text{FBR}/(8 \text{ bits/byte}) \times \text{ATM\_Payload(Bytes/Cell) Cells/s} \tag{2.4}$$

$$P_r = 1/V_r \text{ s/Cell} \tag{2.5}$$

This thesis uses a non-standard ATM_Payload of 64 Bytes/Cell versus the standard 48 Bytes/Cell to ensure commonality with previous protocol analysis work

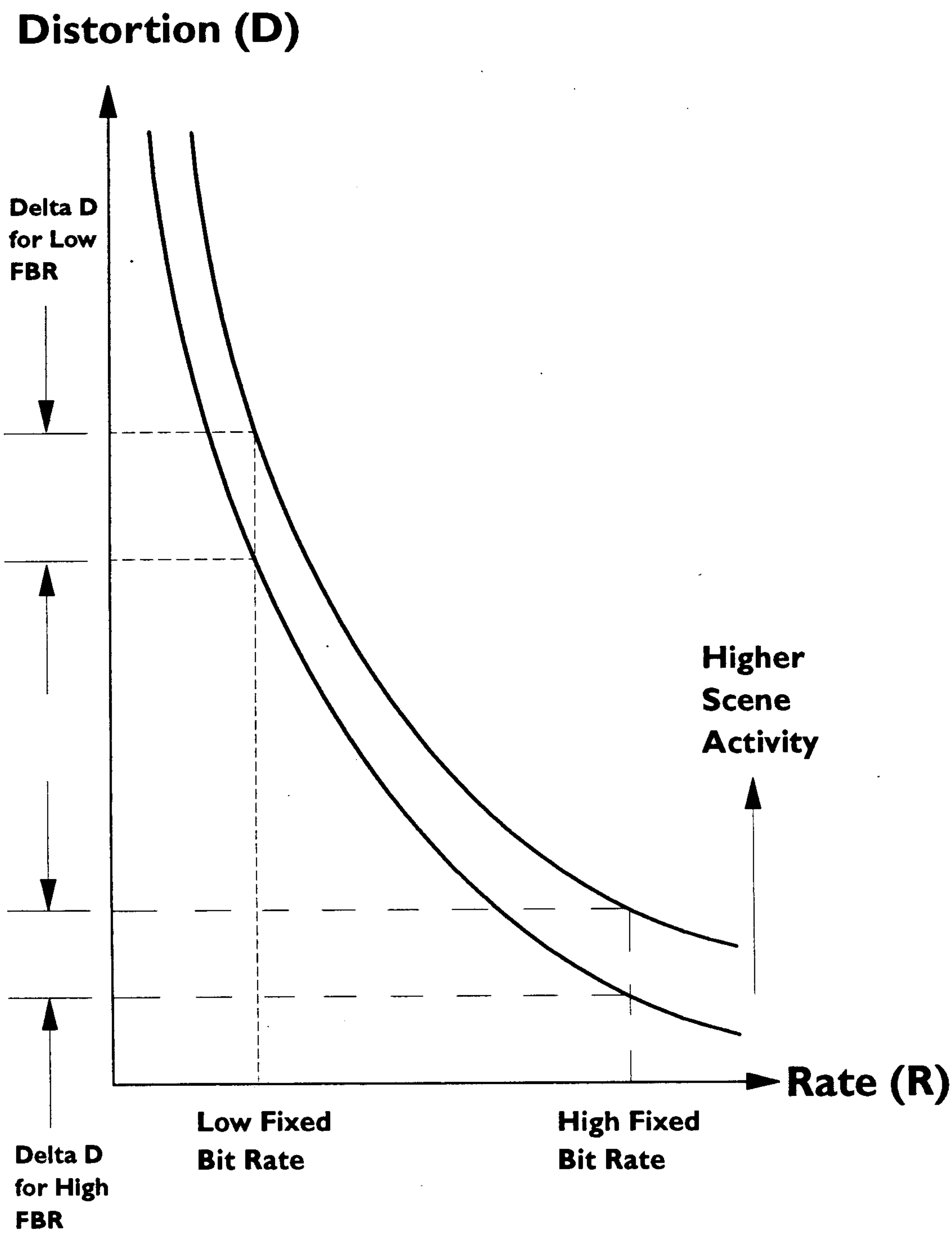

Figure 2.1: Rate-Distortion Function for FBR Video Coding

in [5] (Note: The standard 5 Byte ATM Header is used in this thesis). The P_r for the 45Mbps FBR video source used in this thesis is:

$$V_r = 87{,}890.625 \text{ Cells/sec} \tag{2.6}$$

$$P_r = 11.37778 \ \mu\text{sec/cell} \tag{2.7}$$

2.2 Variable Bit Rate (VBR) versus Fixed Bit Rate (FBR)

VBR works in a way opposite to that of the FBR video sources. To maintain a given constant distortion level, the coding bit rate must vary, thus producing the VBR code. Figure 2.2 illustrates this concept. The VBR encoding introduces a characteristic of the source called burstiness. A measure for indicating burstiness of the video source is defined as follows:

$$B_i = \text{Peak bit rate (PBR) / Average bit rate (ABR)} \tag{2.8}$$

Since PBR equals ABR for a FBR source, the burstiness index B_i for the FBR video source equals one indicating that the video source is not bursty. For a VBR source, the PBR will be higher than the ABR and thus the burstiness index B_i for the a VBR video source will always be greater than one.

To proceed with comparing the DSC protocol performance between FBR and VBR video sources, the computer simulation will recreate the findings found by Ramamurthy (et. al.) [5] for a simulation of a 12 node ATM network (equivalent to a 4 x 4 fully buffered three-stage ATM switch) which uses one FBR video source per link with a total of 4 access links. The performance of the Distributed Source Control with FBR video sources will then be compared to simulation runs with

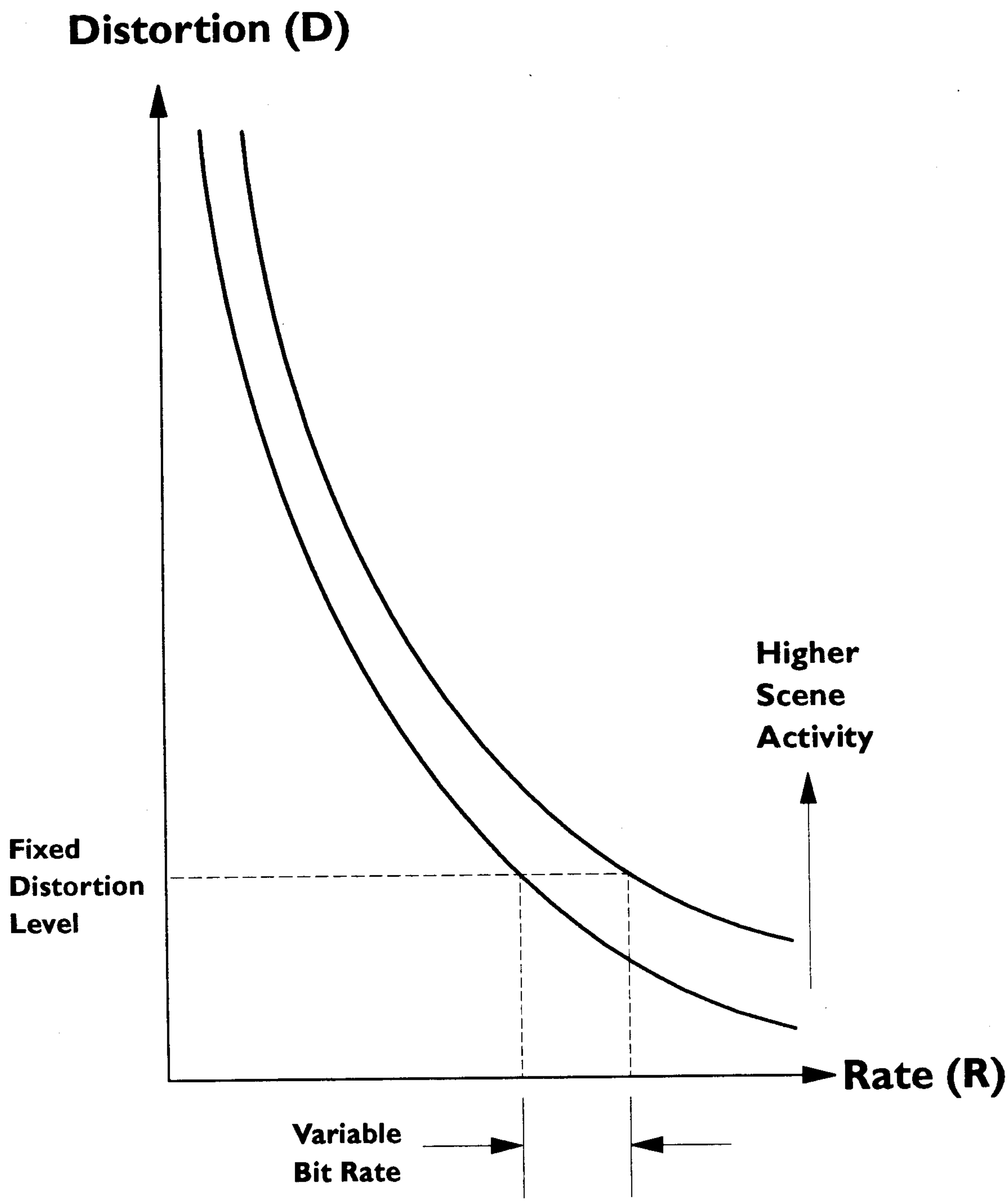

Figure 2.2: Rate-Distortion Function for VBR Video Coding

VBR video sources. The original FBR video sources will be transformed to VBR video sources. The number of VBR video sources and the video source rates will be adjusted as necessary to obtain a statistical equivalent link utilization rate for video.

2.3 VBR Video Source Modeling

To capture accurately the effects of a video source on a VBR encoder, the video source model must take into account the parameters listed in Table 2-1. Video Source Models currently in existence today concentrate primarily on video teleconferencing scenarios that have relatively little movement other than a person speaking into the camera ("talking head"). This restriction in available video source model types limits the simulation from including potential applications that consist of business television, broadcast television including cable, interactive television, and education/training broadcasts. All video teleconferencing models incorporate the effects of the video encoder (codec) used within the video scenario. This type of model either provides the output of the encoder directly or indirectly by having the modeler provide a given time sequence of specific video parameters as the model's input. For the latter case, several types of encoder output are provided by a set of similarly structured video source models.

In addition, a few models will incorporate or accommodate the effects of impulse-like rate changes due to scene changes. However, either a given time sequence of specific video parameters is required to model this effect, or parameters such as actual measured encoder rate changes, probability of scene changes, etc., are required by another. Finally, video rate changes due to camera panning/zooming and special video effects are nearly non-existent within the available set of video source models. If these effects were contained within a video time sequence or encoder output, then a couple of video models could be used to reproduce a similar sequence. However, these effects are transparent to the video source models since they do not produce these effects directly. Rather, they are provided by an input to the model, or by defining certain video rates with a given set of probabilities to recreate similar effects on the encoder output. A comparison of the available video source model options follows.

Table 2-1. Summary of Video Characteristics Contained in Video Models

Video Characteristic	**Occurrence in Video Models**
VBR on a relatively stationary picture (minor movement such as a "talking head" in a video conference application)	Common
Encoder type characteristics on the VBR	Common
Scene changes which produce impulse-like rate changes	Rare
Camera panning	Rare
Camera zooming	Rare
Special video effects (i.e., video window revolving with moving picture inside, advanced fading in/out such as a pattern mosaic, rapidly changing video window size moving around on the video screen)	Rare

2.3.1 Auto Regressive (AR 1) Video Source Model

Maglaris(et.al.) [6] proposed a video source model called a Continuous-State Autoregressive Markov Model that is appropriate for simulation. The video bit rate was obtained from a 10 second video sequence of a talking person in a video teleconferencing scene. A single exponential fit was found to dominate the autocovariance decay. Therefore, only one past value is used in the model, which makes the AR(k) model of order one. This model is used during one of the computer simulation runs to test the Distributed Source Protocol for a bursty video source. The results of this computer simulation are later compared to the original simulation with a FBR video source model. The Continuous-State Autoregressive Markov Model is summarized as follows:

$$\lambda(n) = a\lambda(n\text{-}1) + bw(n) \tag{2.9}$$

where $w(n)$ is a sequence of Gaussian random variables and a and b are constants.

The steady-state average $E(\lambda)$ and discrete autocovariance $C(n)$ are given by:

$$E(\lambda) = [b/(1\text{-}a)] \times \eta \tag{2.10}$$

$$C(n) = [b^2/(1\text{-}a^2)] \times a^n ,\ \ n \geq 0 \tag{2.11}$$

The above model produces a steady-state distribution of λ that is Gaussian with mean $E(\lambda)$ and variance $C(0)$. On the basis of measured data in [6], the video model parameters are defined as:

$$a \cong 0.8781 \qquad b \cong 0.1108 \qquad \eta \cong 0.572 \tag{2.12}$$

and

$$E(\lambda) = 0.52 \text{ bits/pixel} \tag{2.13}$$

$$C(n) \cong 0.0536 \text{ x } (e^{-0.13})^n \text{ (bits/pixel)}^2 \tag{2.14}$$

This AR(1) model involves 2 multiplies and 1 add once the Gaussian random variable is determined. For simulation purposes, this algorithm is efficient, has manageable complexity, does not require much processing power, and allows for accurate control of the link utilization rate for the video sources. As a result, the AR(1) Video Source Model as defined by Maglaris (et. al.) is chosen for one of the VBR video source computer runs.

2.3.2 Auto Regressive (AR 2) Video Source Model

Heyman (et. al.) [7] analyzed several types of video source models. Their analysis used 30 minutes of video teleconference data. This time sequence of video data is 180 times longer than the Maglaris [6] time sequence. The distribution of the number of ATM cells per Frame was found to have the general shape of a negative binomial distribution with mean of 130.2967 and variance of 5536.873.

Due to previous research commonly describing coded video data with an AR(1) model, the AR(k) process has been revisited and investigated in more detail. Using their observed video time sequence data, the AR(1), AR(2), and AR(3) video source models were analyzed for stationarity. Their analysis settled on the AR(2) to resolve correlated residuals in the AR(k) model. The AR(3) model's extra coefficient created a difference with the AR(2) model that was determined to be statistically insignificant, since the standard deviation of a_3 was calculated to be 0.020 and thus close to zero. Their AR(1), AR(2), AR(3) models are summarized as follows:

$$AR(1): X_n = 2.00902 + 0.984594\, X_{n-1} + \epsilon_n \tag{2.15}$$

$$AR(2): X_n = 2.46243 + 1.20682\, X_{n-1} - 0.225705\, X_{n-2} + \epsilon_n \tag{2.16}$$

$$AR(3): X_n = 2.509 + 1.20246\, X_{n-1} - 0.2023\, X_{n-2} + 0.0194\, X_{n-3} + \epsilon_n \tag{2.17}$$

The standard deviation of the residuals is 13.01 for AR(1), which is about 10% of $E(X_n)$, 12.67 for AR(2), and 12.67 for AR(3). Though the standard deviation of the residuals differs, when the AR(1) model is compared to the AR(2) and AR(3) models, the difference is small when compared to $E(X_n)$.

From analyzing 30 minutes of video teleconferencing data, the major conclusions of Heyman (et. al.) [7] were:

1. Number of cells per frame follow a negative binomial (or gamma, the continuous version) distribution
2. Number of cells per frame is a stationary process
3. Autoregressive model of order 2 fits the measured data well overall. However, the model does not produce large enough values of cells for a certain number of video frames to be a good model for traffic studies involving probability of cell loss. This is a key factor, since the occurrence of video frames with a large number of cells may cause additional buffer overflows. As a result, the cell loss probability may be underestimated.
4. A detailed Markov chain model provides enough accuracy that includes producing a large number of cells for certain frames that can be useful in traffic studies

5. A two-state Markov chain model does not provide enough accuracy to be useful in traffic studies. This is due to fitting the mean of the data to the model, which causes a variance value that is too large. Consequently, this model overestimates the loss probability

Because the focus of this thesis is on cell delay and delay variance and not on the probability of cell loss, the autoregressive models defined by Heyman (et. al.) were considered. Among the 3 choices of AR(k) models, we have selected the AR(2) model as recommended in [7] for one of the VBR video source computer runs. This AR(2) model involves 2 multiplies and 3 adds once the Gaussian random variable is determined. For simulation purposes, this algorithm is efficient and manageable in terms of complexity. The model does not require much processing power by itself. However, due to a lower bit rate output, more video sources per ATM link are needed for the computer simulation to keep the same link utilization for video. This aspect of the model increases the overall simulation run time due to managing the additional video sources.

2.3.3 Markov Modulated Poisson Process (MMPP) Video Source Model

Sen (et.al.) [8] proposed a model for an aggregate video source model. Their previous video model captured the state transition rate for a single-activity-level video source model. The extended model consisting of a correlated Markov process model accommodates the multiplexing of statistically different sources (i.e., different means and variances for each source's bit rate). The extended model is very generic and can be *fitted* when each source's mean and variance are known. The main assumption is that the autocovariance behavior of all the sources can be approximated by using the same two dominant time constants. One time constant represents a fast decaying (short term) correlation associated with uniform activity levels and is of the order of a few hundred milliseconds. The other time constant represents a slow decaying (long term) correlation associated with sudden changes in the overall activity level of the scene such as a scene change, or the change between listener and speaker modes in video teleconferencing. The authors used a reduced version of the general extended video model by concentrating on a video teleconferencing application. Three parameters consisting of the mean ratio γ, overall mean λ_{avg}, and the second order statistics in a single activity level $C(\tau)$ completely determine all other required parameters in the video teleconferencing model. The four equations used in matching the measured parameters to the video model are:

1. Conditional autocovariance function $C(\tau)$

$$C(\tau) = C(0)e^{-(a+b)\tau} \tag{2.18}$$

2. Source data rate conditional variance $C(0)$

$$C(0) = Np(1-p)A_l^2 \quad \text{where } p = a/(a+b) \tag{2.19}$$

3. Overall source data rate mean λ_{avg}

$$\lambda_{avg} = NpA_l + qA_h \quad \text{where } q = q/(c+d) \tag{2.20}$$

4. Mean ratio γ is the ratio of the average data rate in the high activity level to the average data rate in the low activity level

$$\gamma = (NpA_l + A_h)/(NpA_l) \tag{2.21}$$

5. The parameter N which indicates the number of quantization levels in any activity level is the only free parameter to be chosen as desired by the modeler

The video teleconferencing application example has the following parameters defined by the authors:

1. Mean data rate per video source $\lambda_{avg} = 3.9$Mbits/s (2.22)
2. Source data rate conditional variance $C(0) = 3.015$ Mbits^2/s^2 (2.23)
3. Short-term correlation exponent $(a+b) = 3.9$/s (2.24)
4. Long-term correlation parameters are chosen to be defined as $c = d$. The authors chose this selection in which the scene in the video teleconferencing example alternates between talking and listening by the same person for approximately equal lengths of time, on the average.

   Note: This assumption is different from the statistics measured for typical voice conversation in which the approximately exponentially distributed talk spurt has a mean $\alpha^{-1} \cong 352$ msec and the approximately exponentially distributed silent period has a mean $\beta^{-1} \cong 650$ msec [9].

5. Average time spent in any one state:

$$1/d = 1/b = 1.5\text{sec} \tag{2.25}$$

$$d = b = 2/3 \ \text{s}^{-1} \tag{2.26}$$

From equation (2.20) and definition (4), $q = 1/2$. From definition (3) and (5), $a \cong 3.2333$ and thus $p = 0.8291$. Depending on the quantization level parameter N chosen by the modeler, A_l and A_h are given by solving equations (2.19), (2.20), and (2.21). The authors did not indicate what value of N was chosen in their analysis.

The author plotted the variation of loss probability with varying buffer sizes for 1 through 5 video sources and three combinations of (utilization percent, mean ration) pairs: (65, 1.5), (65, 2.5), and (75, 1.5).

Once the modeler defines the number of quantization levels for a given video activity level, the video source MMPP model's other parameters can be identified using the specific numbers given by the authors for one video teleconferencing example. For each video frame, a random number is generated to see if any changes occur from the current video activity rate of the source. On the basis of the generated random number, the next state of the video bit rate for the given source is defined as one of the following:

- Stay at the same video bit rate
- Move to a higher bit rate within the same video activity level
- Move to a lower bit rate within the same video activity level
- Move to a lower/higher bit rate of a different video activity level

This model does require additional storage requirements when compared to the autoregressive ones. The amount of storage increases with the number of quantization levels for the video activity levels. However, the processing requirements are significantly lower, since a random number compared to a threshold determines which video bit rate is used within a table. Since the MMPP model was based on the test data from the AR(1) model, the main difference in functionality provided by the MMPP model is the talking and listening modes for the video source. To maintain the same video link utilization rates, this video source model, as defined by the authors, requires additional run time to average out the difference in bit rates between the speaking and listening modes. The computer simulation runs were chosen to be of 2-minute duration to capture the early dynamics of the network, to obtain steady-state numbers for the given network scenario, and to make the processing power required for the simulation to be relatively manageable using a Personal Computer (PC). As a result of the 2-minute duration and the storage limitations, the two-activity-level MMPP video source model defined by the authors is not chosen for this thesis. If the computer run was to be longer than 2 minutes and if the requirement for both speaking and listening modes was necessary, then this model may have been considered a prime candidate.

2.3.4 Bit Rate Predictor of Encoded Video Signals

Rodriguez-Dagnino(et.al.) [10] analyzed several types of video source models (encoder) for the same video time sequence of 46 seconds in duration. Instead of defining a video source model of the bit rate for a given encoder process, a given video time sequence was defined by a set of fundamental parameters (histogram information, spatial correlation, and temporal correlation) which provides the characteristics of the given video time sequence. This video time sequence was then used to predict the bit rate generated by four different types of encoders:

1. Differential Pulse Code Modulation (DPCM)
2. Discrete Cosine Transform (DCT)
3. Subband Coding (SB1)
4. Multidimensional Subband Coding (SBC2)

The bit rate process y_t for the frame t was defined by the authors as follows:

$$y_t = T_t + I_t + N_t \tag{2.27}$$

where T_t is the trend (slow varying component), I_t is the impulsive component, and N_t is the irregular (fast varying component)

The impulsive component I_t was captured by two of the authors' indices: the Vertical Entropy Index and the Temporal Entropy Index and thus were included in most of the linear models. The trend component T_t was captured primarily by three of the authors' indices: the Horizontal Entropy Index, the Variance Index, and the Vertical Entropy Index.

Two sets of fitted models were identified. One set includes the complete set of measured video parameters. A second set was defined by a subset of video parameters (3 parameters only). With a video time sequence including many of the complex video operations (such as scene changes, camera zooming, and camera panning), an ATM Network Protocol can be fully tested to obtain performance measurements. Also, 4 different encoder scheme outputs can be determined from the one video time sequence and, depending upon the network study objectives, one or more of these encoder output types can be used for network simulation using video sources. Upon the research community identifying, measuring, and making available a set of common video time sequences (similar to some of the common still-image pictures used for video compression techniques such as JPEG), the modeling techniques defined in [6] will become very valuable in obtaining indications of network protocol performance in a multimedia environment that can be compared with work from other authors.

The 3-parameter video source models defined by the authors were reviewed very closely for our own simulation. Ideally, these models provide much flexibility in testing out the effects of different video encoder's output bit rate on a network protocol. Also, the 3-parameter models consisting of 3 multiplies and 3 adds per frame are only one more multiply higher than the AR(2) video source model.

However, these video models can not be used without obtaining the time sequence of the various indices defined in [10] for the video sequence under consideration.

2.4 Video Source Models Selected for Use in Computer Simulation

With all the pros and cons addressed in the previous paragraphs of various options for VBR video source modeling that produces bursty data, we now utilize two models for measuring the performance of the Distributed Source Control Protocol:

1. AR(1) as defined by Maglaris (et.al.) [6]
2. AR(2) as defined by Heyman (et.al.) [7]

The other two video models considered from [8] and [10] were not chosen for the network simulation. The primary reason being the equivalent test equipment setup, which was not readily available to obtain a video time sequence that can fully utilize the advantages of these models. Also, a subset of the MMPP video model was not chosen due to the requirement of additional run time to average out the difference in bit rates between the speaking and listening modes. Since the speaking and listening modes as a pair were not a requirement and the MMPP model was based on the test data from the AR(1) model, the talking mode of the MMPP model is still retained by the selection of the AR(1) model.

The AR(1) and AR(2) models were chosen for the PC network simulation due to their manageable complexity for a 2-minute computer run, minimal storage requirements, accurate control of the link utilization rate for the video sources, and their accuracy for a network simulation study concentrating primarily on cell delay. The AR(1) model produces more cells per video frame than the AR(2) model and thus, requires fewer video sources per link to obtain the link utilization rate goal. The AR(2) with more sources per link will benefit from effects of statistical multiplexing.

Since the AR(2) model is based on a 30 minute test sequence while the AR(1) model is based on a 10 second test sequence, results of the network's DSC protocol subjected to bursty video data from these two video source models will be compared to each other. Also, the results using the bursty video sources will be compared to the FBR video model defined in [5] and Section 2.1.

CHAPTER 3

ADAPTIVE DISTRIBUTED SOURCE CONTROL

This chapter presents an adaptive Distributed Source Control (DSC) control algorithm for a Source Node along with a non-adaptive DSC control algorithm. The other network nodes (access, intermediate, and destination nodes) provide feedback on certain network conditions such as potential queue overflow and average delay through the queues. For a given control algorithm, the probability distributions can be found or defined to optimize the control algorithm results. Even with general estimates of distributions for a given data type, unless an algorithm adjusts according to the specific distribution of the data type being transmitted, the control algorithm will produce sub optimal results. By identifying an adaptive DSC algorithm, the algorithm may be seeded with default general estimates of distributions for a given data type or seeded by the application input of these data distributions. As the source transmission proceeds, the control algorithm is allowed to adapt to the specific data's actual distribution characteristics within some pre-defined boundaries. The proposed algorithm produces results approaching a more optimal solution than the algorithm without the adaptive characteristic. The adaptive DSC algorithm expands the original work performed in [5] for a non-adaptive DSC algorithm that focused on non real-time data. See Section 1.2.1.1 for an overview of the non-adaptive DSC algorithm presented in [5].

3.1 Adaptive DSC Control for a Source Node

The objective of the adaptive DSC for a source node is to avoid congestion and to avoid dropping cells, whenever possible. The adaptive DSC network management scheme is a mix of a network edge and network (internal) management. Also, the adaptive DSC scheme is a hybrid rate control and policing function. To achieve the objectives of the adaptive DSC scheme, the implementation includes a DSC control parameter *W/T*, which represents *W* cells per *T* smoothing time constant (milliseconds), as the primary control mechanism. The effect of the adaptive DSC's *W/T* control parameter is to reshape the distribution of a bursty source's cell transmission rate. Instead of a high concentration of back-to-back cell transmission followed by periods of inactivity, the higher back-to-back cell transmission rate is averaged over time to spread out some of the back-to-back cell transmissions to the inactivity time periods. Quality of Service (QOS) for real-time sources can be improved by the use of the DSC scheme. The DSC scheme improves the QOS by assisting in the transmission of cells prior to the source buffer receiving the next batch of cells (i.e., a set of cells representing a video frame). The source model monitors each basic data type:

- Data
- Video
- Audio

The source is structured in such a way that each one of the basic data types has separate transmit buffers. This requirement is necessary for optimizing the DSC to each source's unique characteristics. Alternatively, grouping all the source data types into one buffer can be used if data type priority is used within the single buffer. This approach leads to a suboptimal approach since the DSC decisions are based on the one buffer size. As a minimum, data type priority provides a criterion to base decisions regarding which cell(s) need to be discarded upon source buffer overflow or network congestion. Now if the suboptimal approach has the priority scheme removed, the single buffer will be a major weak point in network performance for quality of service compliance. This results from the fact that certain data types will be indiscriminately discarded without regard to other data types further within the buffer. Other types of data may tolerate a delay of retransmission, such as a data file when discarded versus other time-critical data types.

3.1.1 Control Parameters

The adaptive DSC control algorithm in this paper monitors the following parameters to manage and shape the bursty source's cell transmission rate:

1. Data_Type_Queue_Size
   - Average
   - Increasing/Decreasing queue size indicator
2. Data_Type_Queue_Overflow
3. DSC_Time_Constant
4. Average_Cell_Transmission_Rate

The adaptive DSC control algorithm obtains feedback from the network on the average end-to-end network delay and on any potential network queue overflow conditions along a virtual call's path by the following parameters:

1. Average_End_to_End_Network_Delay
2. Network_Queue_Overflow_Warning

Other parameters can be included into the adaptive DSC design based on the network and QOS objectives.

3.1.2 DSC Source Queue Average Delay Monitoring

The adaptive DSC algorithm relies on the Source Queue Average Delay for the majority of its source queue overflow avoidance. This algorithm obtains a measurement of the source's average output rate. For adjusting the DSC control parameters, when the queue average delay exceeds certain thresholds, a high and low threshold are maintained for each data type involved with a call. If the value of the Queue_Average_Size[data_type] parameter is greater than the threshold value $\text{Queue\_Average\_Size\_threshold}_{\text{High}}$ or lower than the threshold value $\text{Queue\_Average\_Size\_threshold}_{\text{Low}}$ defined at call set up, then the DSC control parameters are adjusted significantly. The $\text{Queue\_Average\_Size\_threshold}_{\text{High}}$ algorithm is defined below for the case when the threshold is exceeded:

Algorithm (High Threshold):

1. Increase the DSC Window for the Source

$$W_S = W_S^{+} \tag{3.1}$$

2. Increase the average transmit rate

$$\lambda^{+}_{\text{avg}} = W_S^{+}/TC_{\text{DSC}} \tag{3.2}$$

3. Request acceptance of the average transmit rate by the network
   - Forward request to the next network node
   - Each node examines rate request and either
     - Accepts requested rate and reserves bandwidth and buffers at the node, or

– Marks down requested rate based on bandwidth availability at node

- The destination node returns the final requested transmit rate λ^{+}_{Net} to the source node

4. The Source either
   - Accepts λ^{+}_{Net} and continues to Step 5
   - Rejects and proceeds with monitoring other source data types involved in a call

5. The following parameter is reset
   - Potential_Mark_Down$_{(i)}$ is reset to FALSE (Generated by the Algorithm for checking the low threshold)

6. Source transmits with new DSC average rate

$$\lambda = \lambda^{+}_{Net} \tag{3.3}$$

7. Monitor the queue size average condition for each of the source's data types involved in the call and repeat steps 1 through 6 upon detecting a high queue size condition

The Queue_Average_Size_threshold$_{Low}$ algorithm is defined below for the case when the threshold is not exceeded:

Algorithm (Low Threshold):

The queue size average for a given data type is compared to the Queue_Average_Size_threshold$_{Low}$. If the queue size average is below the threshold, the following algorithm is performed.

1. Check to see if given data type has been marked for potential DSC throttle down
    - If not, proceed with step (2)
    - If already marked, proceed with checking the queue parameters of the next data type involved with a call from the given source
2. Check to see if given data type has been *guaranteed* by the network
    - If not, proceed with step (3)
    - If guaranteed, proceed with checking the queue parameters of the next data type involved with a call from the given source
3. Mark $data\_type_i$ for potential DSC throttle down

$$M_i = \text{TRUE} \tag{3.4}$$

5. The given data type proceeds to the Network Congestion Monitoring in Section 3.1.4.
6. Monitor the queue size average condition for each of the source's data types involved in the call and proceed to the Network Congestion Monitoring algorithm
    - Repeat steps 1 through 5 upon detecting a low queue size condition

3.1.3 DSC Source Queue Overflow

The Queue Size Average Monitoring algorithm should prevent most queue overflows and thus, the adaptive DSC algorithm does not rely on DSC Queue Overflow protection as its main method for congestion avoidance. However, for the case in which the data type transmission needs exceed the Queue Size Average Monitoring algorithm, the DSC Queue Overflow algorithm initiates.

One approach for adjusting the DSC control parameters, when overflow occurs at the source queue, is to maintain an overflow count parameter for each data type involved with a call. If the DSC queue overflow rate indicator

Queue_Overflow_Count[data_type]/Cell_Sent_Count[data_type]

is greater than a Queue_overflow_threshold[data_type] defined at call set up, then the DSC control parameters are adjusted. Since the occurrence of queue overflow commonly is associated with a series of consecutive overflow conditions, a second approach consists of the adaptive control being initiated for queue overflows upon the occurrence of the first queue overflow. Rather than waiting for the first queue overflow, the DSC Queue Overflow algorithm is initiated when the queue size reaches a certain size (50-80% of the queue size for example). The computer simulation models this latter approach. The queue size threshold should vary according to the source buffer input rate. For the simulation, the queue size threshold is set at a fixed percentage.

Algorithm:

1. Decrease the DSC Window for the Source

$$W_S = W_S^- = W_S\,(\text{old})/2 \tag{3.5}$$

2. Decrease the DSC time constant for the source by the same percentage

$$TC_{\text{DSC}} = TC_{\text{DSC}}^- = TC_{\text{DSC}}(\text{old})/2 \tag{3.6}$$

3. Maintain the same average transmit rate and adjust for any round off errors

$$\lambda_{\text{avg}} = W_S^-/TC_{\text{DSC}}^- = W_S(\text{old})/TC_{\text{DSC}}(\text{old}) \tag{3.7}$$

4. Request acceptance of the average transmit rate with the new W and TC_{DSC} parameters by the network (Note: The rate due to rounding may be slightly higher than previously stated so each node may need to perform slight adjustments. This step can be bypassed if the ratios are exact with the new W and TC parameters)
    - Forward request to the next network node
    - Each node examines rate request and either
        - –Accepts requested rate and reserves bandwidth and buffers at the node, or
        - –Marks down requested rate based on bandwidth availability at node
    - The destination node returns the final requested transmit rate λ_{Net} to the source node via the access node
5. The Source either
    - Accepts λ_{Net} and continues to Step 6
    - Rejects and proceeds with monitoring other source data types involved in a call
6. The time constant counter for the data type $TC_{\mathrm{DSC}(i)}$ is reset
7. Source transmits with new DSC average rate

$$\lambda = \lambda_{\mathrm{Net}} \tag{3.8}$$

8. Monitor the queue overflow condition for each of the source's data types involved in the call and repeat steps 1 through 7 upon detecting a queue overflow

3.1.4 Network Congestion Monitoring

When the network informs the source of a network bottleneck, the data types marked down for potential throttle down in Section 3.1.2 are adjusted for a lower transmission rate.

The Network Congestion Monitoring algorithm is defined below:

Algorithm:

1. Check to see if the DSC *W/T* parameter for a given data type has been reduced to a minimum value $W_{\mathrm{min}}/TC_{\mathrm{DSC}}$ or $W/TC_{\mathrm{DSC(min)}}$
   - If the DSC W/T parameter is at a minimum value, proceed with step (2)
   - If not, decrease the following
     – DSC Window for the source
     $$W_S = W_S^- = W_S\,(\mathrm{old})/2 \tag{3.9}$$
     – DSC Time Constant for the source
     $$TC_{\mathrm{DSC}} = TC_{\mathrm{DSC}}^- = TC_{\mathrm{DSC}}(\mathrm{old})/2 \tag{3.10}$$
     while maintaining the same average transmission rate
     $$\lambda_{\mathrm{avg}} = W_S^-/TC_{\mathrm{DSC}}^- = W_S(\mathrm{old})/TC_{\mathrm{DSC}}(\mathrm{old}) \tag{3.11}$$
2. Check to see if given data type has been throttled down
   - If not, proceed with step (3)
   - If already throttled down, proceed with checking the throttle down condition of the next data type involved with a call from the given source
3. Check to see if all data types have been throttled down
   - If not, proceed with step (4)

- If all data types have already been throttled down, proceed with throttling down lowest priority data type (or a Data source if priority information is unavailable)
    - Proceed with step (4)

4. Start the throttle down process for the data type by decreasing the DSC Window for the Source

$$W_S = W_S^- \tag{3.12}$$

5. Decrease the average transmit rate

$$\lambda^-_{\text{avg}} = W_S^- / TC_{\text{DSC}} \tag{3.13}$$

6. Forward new lower average transmit rate to the network

7. The following parameters are reset
    - The time constant counter for the data type $TC_{\text{DSC}(i)}$ is reset
    - Potential_Mark_Down$_{(i)}$ is reset to FALSE (Generated by the Algorithm for checking the queue's average size low threshold)

8. Source transmits with new DSC average rate

$$\lambda = \lambda^-_{\text{avg}} \tag{3.14}$$

9. Monitor both of the following
    - Network congestion condition
    - The marked data type for potential throttle down during network congestion

    for each of the source's data types involved in the call

10. Repeat steps 1 through 9 upon detecting a Network congestion condition

3.1.5 Network Queue Average Delay Monitoring and Tracking

The adaptive DSC algorithm relies on the Network Queue Average Delay Monitoring and Tracking for the majority of its network QOS maintenance. This algorithm is similar to that listed in Section 3.1.2. This algorithm has an advanced feature which tracks and adjusts the DSC control parameters according to periodic feedback from the network. For adjusting the DSC control parameters when the queue average delay exceeds certain thresholds, a high and low threshold are maintained for each data type involved with a call. The average network queue delay high and low time values are converted to an equivalent queue size based on the source's cell service time (3.68 μsec in the computer simulation). If the value of the Network_Queue_Average_Size[data_type] parameter is greater than a threshold Network_Queue_Average_Size_threshold$_{High}$ or lower than a threshold Network_Queue_Average_Size_threshold$_{Low}$ defined at call set up, then the DSC control parameters are adjusted significantly.

The adaptive DSC algorithm provides some fine tuning of the DSC control parameters by dividing the high and low threshold area into quadrants. Also, a region just outside the high (+5%) and low (-10%) thresholds are defined to assist in the fine tuning of the parameters. A set of scalars is used to scale the DSC parameters based on which region the network queue average resides and which direction (higher or lower) the average is heading.

The Network_Queue_Average_Size_threshold algorithm for both the high and low thresholds is defined below for the case when the threshold is exceeded:

Algorithm (Decrease = High Threshold; Increase = Low Threshold):

1. Decrease (Increase) the DSC Window for the Source:

$$W_S = W_S^- = W_S \text{ (old) x scalar[region, avg\_direction]} \qquad (3.15)$$

2. Decrease (Increase) DSC Time Constant for the source:

$$TC_{\mathrm{DSC}} = TC_{\mathrm{DSC}}^{-}$$
$$= TC_{\mathrm{DSC}}(\mathrm{old}) \times \mathrm{scalar[region, avg\_direction]} \quad (3.16)$$

3. Maintain the same average transmission rate:

$$\lambda_{\mathrm{avg}} = W_S^{-}/TC_{\mathrm{DSC}}^{-} = W_S(\mathrm{old})/TC_{\mathrm{DSC}}(\mathrm{old}) \quad (3.17)$$

4. Request acceptance of the average transmit rate by the network

- Forward request to the next network node
- Each node examines rate request and either
  - –Accepts requested rate and reserves bandwidth and buffers at the node, or
  - –Marks down requested rate based on bandwidth availability at node
- The destination node returns the final requested transmit rate λ_{Net} to the source node

5. The Source either

- Accepts λ_{Net} and continues to Step 6
- Rejects and proceeds with monitoring other source data types involved in a call

6. The following parameters are reset

- The time constant counter for the data type $TC_{\mathrm{DSC}(i)}$ is reset

7. Source transmits with new DSC average rate

$$\lambda = \lambda_{\mathrm{Net}} \quad (3.18)$$

8. Monitor the queue size average condition for each of the source's data types involved in the call and repeat steps 1 through 7 upon detecting a high queue size condition

The response to the quadrant thresholds being exceeded is similar to the actions taken for the high and low thresholds defined above.

3.2 Network Distributed Source Control Algorithm

The adaptive DSC algorithm presented in Section 3.1 is initiated at the time a source call request has been accepted by the network. In this section, the DSC control algorithm used by the network edge and intermediate nodes is presented in detail. The DSC control algorithm defined in this chapter is based upon the DSC control algorithm proposed by Ramamurthy (et. al.) [5]. The modified DSC control algorithm specifically addresses different data types along with integrating the adaptive DSC control algorithm protocol of the source.

3.2.1 DSC Call Setup Procedure

Each Source node requests a DSC average transmit rate from the network access node for each of its data types involved in a call. This request initiates the Call Setup Procedure defined as follows:

Algorithm:

1. Source data type requests an average transmit rate that is valid when the source is in an active state
   - $\lambda_{requested} = \lambda_{avg}$ (3.19)
   - minimum acceptable average transmit rate = λ_{min} (3.20) (optional)
2. Source's request for acceptance of the average transmit rate is forwarded by each node of the network
   - Each node examines rate request and either

– Accepts requested rate

– Marks down requested rate based on bandwidth availability at node

- Each node reserves bandwidth and buffers according to the network's acceptable requested source rate
- Forward resulting source rate request value to the next network node if DSC window size is non-zero

– If DSC window is zero, the intermediate node must inform the previous node to look for another path. This process continues until either a path is found or the network determines that the call is blocked.

- The destination node returns the final requested transmit rate λ_{Net} to the source node

4. The Source either

- Accepts λ_{Net} (Call Acceptance) and continues to Section 3.2.2
- Rejects λ_{Net} and call is blocked. Source must retry call by returning to its adaptive DSC control algorithm to determine the new requested rate and returns to step (1) of this algorithm (Section 3.2.1).

3.2.2 Initiating the Adaptive DSC Source Algorithm

The Adaptive DSC Algorithm for the source as defined in Section 3.1 is initiated after either one of the following conditions: 1) acceptance of the call, or 2) blocked call. The acceptance or blocking of the call occurs in Step 4 of Section

3.2.1. In parallel with the adaptive DSC algorithm for the source, an adaptive end-to-end DSC algorithm is initiated as described in Section 3.2.3.

3.2.3 Adaptive End-to-End DSC Control

This control is a secondary control mechanism for the overall DSC control strategy. The Network DSC control and the adaptive DSC control algorithm for the source are network congestion avoidance methods. The adaptive end-to-end DSC control is a reactive control mechanism. The adaptive end-to-end DSC control algorithm runs continuously within the network.

Algorithm:

1. At each network node, the End-to-End DSC Control monitors the following for congestion
   - buffer overflow, or
   - check percentage of buffer full (Note: this technique is used for the computer simulation)
2. Each network determines if congestion is detected
   - If no detection of congestion, the node returns to step (1)
   - If node detects congestion, then proceed to step (3)
3. Decrease the DSC End-to-End Window for the network

   $$W_E = W_E^- = W_E(\text{old})/2 \tag{3.21}$$

4. Send reduced W_E information to affected access node
5. Indicate network congestion information to the source
   - Network Congestion Indicator (or potential congestion warning)

- Data Type
- Call

Note: The source uses this information for its DSC Queue Overflow Algorithm described in Section 3.1.3.

6. Monitor network congestion over a time constant

$T_{Congest}$

$(T_{Congest(min)} = \text{Network's Cell Service Time})$ (3.22)

7. Each network along congested path determines if congestion is removed

   - If congestion has not been removed, the node checks to see if $T_{Congest}$ has expired
     - If $T_{Congest}$ has not expired, return to Step (6)
     - If $T_{Congest}$ has expired, return to Step (3)
   - If congestion has been removed, the node checks to see if $T_{Congest}$ has expired
     - If $T_{Congest}$ has not expired, return to Step (6)
     - If $T_{Congest}$ has expired, check overflow percentage against a congestion count threshold $C_{threshold}$
       - If lower than threshold, proceed to Step (8)
       - If higher than threshold, reset $T_{Congest}$ and return to step (6)

8. The network node requests to increase the Network's DSC End-to-End Window by forwarding request to next network node

$$W_E = W_E^+ \tag{3.23}$$

9. Request acceptance of the DSC End-to-End Window by the network

- Each node examines window request and either
  - Accepts requested End-to-End Window; reserves bandwidth and buffer at the node
  - Marks down requested window based on buffer availability at node
- The destination node returns the requested DSC End-to-End Window to the access node
- The access node forwards the request to the next node within the network

10. The requesting node either
    - Accepts $W_E{}^{+}\mathrm{Net}$ and continues to Step 1
    - Rejects and continues to request original W_E

CHAPTER 4

COMPARISON OF VIDEO SOURCE MODELS AND ADAPTIVE/NON-ADAPTIVE DSC ALGORITHMS

In this chapter, the video sources selected for simulation will be reviewed based on the simulation results. Also, the performance of the adaptive DSC algorithm will be contrasted with that of the non-adaptive DSC algorithm.

4.1 Non-Adaptive DSC Algorithm Analysis

The non-adaptive DSC algorithm is used for 3 basic network scenarios:

- Fixed Bit Rate Video Source
- Variable Bit Rate Video Source AR(1)
- Variable Bit Rate Video Source AR(2)

For each one of these network scenarios, a bursty data source is used with a non-adaptive DSC control parameter. For each one of the 3 basic network scenarios, two network simulation computer runs are performed with one case at a high DSC control parameter (320 cells/50 msec for data and 390 cells/50msec for video) and another case at a low DSC control parameter (8 cells/1.25 msec for data and 10 cells/1.25 msec for video).

4.1.1 Fixed Bit Rate Video Source Analysis

The FBR video source scenario, which includes fixed bit rate audio along with bursty data sources, had the best performance among all computer simulations. This is due to just having one source type that was bursty. This scenario was used as a baseline for comparing the effects of the fixed bit rate sources conversion to bursty video sources. Between the two fixed DSC control settings for W/T (W cells per time constant T), the lower W/T setting improved the network performance significantly in terms of delay, delay variance, and percentage of cells with delay greater than 1 msec. These results are illustrated in the figures within this chapter (Delay, Delay Variance, and Percentile > 1msec graphs).

With the W/T parameters set at $W = 320$ cells, $T = 50$ msec, the FBR video source scenario had delays of 819 μsec, 582 μsec, and 572 μsec for data, video, and audio (voice), respectively. As the W/T parameters are lowered to $W = 8$ cells, $T = 1.25$ msec, the second FBR video source scenario had delays of 29 μsec, 21 μsec, and 22 μsec for data, video, and audio (voice), respectively. This represents a 96% reduction in the mean waiting time for all 3 source types. For both settings of the DSC W/T control parameter, the FBR video source scenario achieved a steady-state condition approximately 95 to 100 seconds into the simulation.

For the delay variance, which represents an indication to the average delay jitter as seen by the each node through the network, the FBR video source scenario had delay variances of 0.880 μsec^2, 0.770 μsec^2, and 0.305 μsec^2 for data, video, and audio (voice), respectively, with the W/T parameters set at $W = 320$ cells, $T = 50$ msec. As the W/T parameters are lowered to $W = 8$ cells, $T = 1.25$ msec, the second FBR video source scenario had delay variances of .001 μsec^2, .001 μsec^2, and .001 μsec^2 for data, video, and audio (voice), respectively. This represents a

99.9% reduction for data and video and a 99.7% reduction for audio in the mean waiting time variance.

The percentile of cells with delays greater than 1 msec is a performance measure that identifies a potential set of cells that may require eventual discarding if cells are related to a real-time source, and the network contains many network hops between the source and destination nodes. When focusing on cell delays greater than 1 msec, the FBR video source scenario had percentiles of 0.3049, 0.2155, and 0.2123 for data, video, and audio (voice), respectively, with the W/T parameters set at $W = 320$ cells, $T = 50$ msec. As the W/T parameters are lowered to $W = 8$ cells, $T = 1.25$ msec, the second FBR video source scenario had cell percentiles with delays greater than 1 msec near 0 (zero) for all 3 source types. This represents a reduction of almost 100%.

4.1.2 Variable Bit Rate Video Source Analysis

The VBR video source analysis includes two types of bursty video scenarios. One analysis includes the video source as an AR(1) model. This model produces the higher bit rate per source for the video data. The other VBR video source analysis includes the video source as an AR(2) model. In terms of average cells per video frame used in the simulation, the AR(1) model produces approximately 244.1406 cells per frame while the AR(2) model produces approximately 100.7124 cells per frame. These numbers are valid when the ATM cell is defined as 69 Bytes (64 Byte payload and 5 Byte header) and each source is acting as a NTSC video source. Both scenarios include the fixed bit rate audio source model along with the bursty data type as in Section 4.1.1.

4.1.2.1 Variable Bit Rate Video Source AR(1) Model Analysis

In this network scenario, the 45Mbps FBR video source on each 150Mbps ATM link in Section 4.1.1 is replaced with 12 VBR AR(1) video sources. The average bit rate of the AR(1) model in [6] was adjusted from 0.52 to 0.50 bits per pixel to obtain the same effective link utilization rate of 32.34% used in Section 4.1.1. The average cells per frame generated by these sources is 244 cells per frame at the NTSC rate (30 frames per second). This scenario represents 12 video sources on each of the 4 links. Each video source represents a video teleconferencing site in a speaking mode while each of the destination sites is considered to be in a listening mode during the 2 minute simulation.

For the higher *W/T* control, the delay average has a slight steady increase between 20 to 95 seconds into the simulation. After 95 seconds, the delay values level out for the remainder of the simulation. The delays for all 3 data types are higher than the similar case in Section 4.1.1. In contrast to the case in Section 4.1.1, the delay values for the bursty video sources track closer to those of the bursty data sources instead of the FBR audio sources. Similar to the VBR delay results, both the delay variance and the percentile of cells with delays greater than 1 msec start leveling off during the 95 to 120 seconds of the simulation. Both of these measured parameters are higher than the case in Section 4.1.1.

For the lower DSC *W/T* parameter, the delay average hits a high point around 95 seconds into the simulation and levels off for the remainder of the simulation. Around 95 seconds into the simulation, both the delay variance and the percentile greater than 1 msec values have a large step increase and then level off. This phenomenon at the lower level happens later in the simulation than in Section 4.1.1 and others. Of all the lower DSC value simulations, this characteristic happens the latest into the simulation thus indicating a longer time to reach steady state and is intuitively consistent with adding an additional bursty source type when compared to Section 4.1.1.

With the W/T parameters set at $W = 320$ cells, $T = 50$ msec, the VBR video source AR(1) scenario had delays of 1295 μsec, 1340 μsec, and 1059 μsec for data, video, and audio (voice), respectively. As the W/T parameters are lowered to $W = 8$ cells, $T = 1.25$ msec, the second VBR video source AR(1) scenario had delays of 42 μsec, 40 μsec, and 35 μsec for data, video, and audio (voice), respectively. This represents a 97% reduction in the mean waiting time for all 3 source types. For the lower DSC W/T parameter, all three network performance parameters are higher than those from Section 4.1.1 However, these parameters approach the values from Section 4.1.1. In terms of absolute values, these measured values are very close to the bursty data only scenario. The delay increase is from 29 μsec to 42 μsec for data (45% increase), 21 μsec to 40 μsec for video (90% increase), and 22 μsec to 35 μsec for audio (59% increase). The video source had the most significant change in delay performance as expected since the source was changed from a non-bursty source (FBR) to a bursty source (VBR) in this network scenario.

For the delay variance, the VBR AR(1) video source scenario had delay variances of 2.028 μsec^2, 1.982 μsec^2, and 1.922 μsec^2 for data, video, and audio (voice), respectively, with the W/T parameters set at $W = 320$ cells, $T = 50$ msec. As the W/T parameters are lowered to $W = 8$ cells, $T = 1.25$ msec, the second scenario for the VBR AR(1) video source had delay variances of .004 μsec^2, .003 μsec^2, and .003 μsec^2 for data, video, and audio (voice), respectively. This represents a 99.8% reduction for all 3 source types in the mean waiting time variance.

For the case of cell percentiles with delays greater than 1 msec, the VBR AR(1) video source scenario had cell percentiles of 0.4476, 0.4697, and 0.3677 for data, video, and audio (voice), respectively, with the *W*/*T* parameters set at $W = 320$ cells, $T = 50$ msec. As the *W*/*T* parameters are lowered to $W = 8$ cells, $T = 1.25$ msec, the second scenario for the VBR had cell percentiles with delays greater than 1 msec very near to 0 (zero) for all 3 source types with the values being 0.0003, 0.0002, and 0.0002 for data, video, and audio, respectively. This represents a reduction of 99.9% for all 3 source types.

4.1.2.2 Variable Bit Rate Video Source AR(2) Model Analysis

In this network scenario, the 45Mbps FBR video source on each 150Mbps ATM link in Section 4.1.1 is replaced with 29 VBR AR(2) video sources. The cell rate of the AR(2) model in [7] was adjusted from an average of 130.2967 cells per frame at the PAL rate of 25 frames/sec. The adjustment resulted in a cell rate of 108.5805 cells per frame at the NTSC rate of 30 frames/sec. In addition, the 108.5805 cells per frame were adjusted further to 100.71 cells per frame to adjust for the ATM cell size (64 Bytes versus 69 Bytes used in the simulation). The adjusted cell rate for the video sources in this network scenario was to provide the same effective link utilization rate of 32.34%, approximately, used in Section 4.1.1. However, a 1% increase of approximately 33.34% occurred due to the AR(2) model needing to proceed into negative values to provide the expected average. Instead, the simulation clipped any negative values to 1 cell per frame minimum resulting in the approximate 33.34% link payload average for video. This scenario represents 29 video sources on each of the 4 links and each video source represents a video teleconferencing site in a speaking mode while each of the destination sites is considered to be in a listening mode during the 2 minute simulation.

For the higher *W*/*T* control, the delay average is a fairly steady throughout the simulation with a slight dip around the 60 second time period. Otherwise the delay average is stable after the 25 second period. The fixed DSC W/T parameter used for the AR(2) case was the same identical W/T value as the AR(1). For the AR(2) case, each source has a lower average bit rate so the W/T value reserves more bandwidth of the network. However, on average this extra bandwidth is not used. For cases of large cells per frame, the AR(2) case fairs better in this case than the AR(1) video source since the AR(2) has more free token cells to use per frame. When these free tokens become available, the AR(2) can take advantage of the cells and use any additional cells if available. This leads to the fact that unless the actual source statistics are known ahead of time, the DSC W/T parameter may or may not be optimal. The delay average, variance, and percentile of cells greater than 1 msec values have similar shapes over the plotted time period . The graphs have a dip in the middle of the simulation at 60 seconds and a slight upward bow shape on either side. After 90 seconds, the various plots level out during the 90 to 120 seconds into the simulation except for the delay variance, which has a slight decrease.

For the lower DSC W/T parameter, all three network performance parameters are higher than those from Section 4.1.1 and that of Section 4.2.1. The values are close to the Section 4.2.1 number initially between the 5 to 30 second period of the simulation. However, at the 30 second mark, all 3 performance parameters have a noticeable increase. This effect is attributed to the slow and steady increase of the bursty data source from the 5 through 40 second mark. The other two scenarios at the low DSC value have the data source payload average decreasing prior to the 40 second mark (15 and 20 seconds, respectively). All of the AR(2) video source parameters peak at the 30 second mark. The percentile of cells greater than 1 msec value steadily decreases until the 90 second mark while the delay average and delay variance start with a slight increase around the 105 second mark before returning to a leveling or decreasing value mode. Some of these numbers are quite small so a slight increase in cell delay can result in a noticeable change in the graphs.

The VBR video source AR(2) scenario had delays of 1048 μsec, 899 μsec, and 797 μsec for data, video, and audio (voice), respectively, with the *W*/*T* parameters set at $W = 320$ cells, $T = 50$ msec. As the *W*/*T* parameters are lowered to $W = 8$ cells, $T = 1.25$ msec, the second VBR video source AR(2) scenario had delays of 51 μsec, 50 μsec, and 43 μsec for data, video, and audio (voice), respectively. This represents a 95% reduction, approximately, in the mean waiting time for all 3 source types.

For the lower DSC *W*/*T* parameter, all three network performance parameters are higher than those from Section 4.1.1 and Section 4.2.1. The increase in these values when compared to those of Section 4.1.1 is due to the conversion of the one FBR video source to 29 VBR video sources. As to the increase above the delay values obtained from Section 4.2.1 for the lower DSC *W*/*T* parameter, the fact that the percentage reduction was less for the AR(2) video source case than the AR(1) case is an indicator of the mismatched DSC *W*/*T* parameter for the AR(2) video source. Another contributing factor at the lower DSC *W*/*T* parameter is the 1% increase in the link payload attributed to the AR(2) video source model. For the higher DSC *W*/*T* parameter case, the lower delay values from those of the AR(1) model are attributed in part by the additional video sources per link while maintaining the same link payload rate. The time constant is longer than the frame rate of the video sources allowing the cells per frame to build up and finish on average prior to the arrival of the next batch of available free cell tokens represented by the *W* parameter. A higher level of statistical multiplexing represented by the uniformly distributed 29 video sources on the link versus the 12 video sources as in Section 4.2.1 along with the combination of the DSC *W*/*T* parameter having a longer time constant provide a form of traffic smoothing not available for Section 4.2.1 at this DSC setting.

For the delay variance, the VBR AR(2) video source scenario had delay variances of 1.299 μsec^2, 1.182 μsec^2, and 1.179 μsec^2 for data, video, and audio (voice), respectively, with the W/T parameters set at $W = 320$ cells, $T = 50$ msec. As the W/T parameters are lowered to $W = 8$ cells, $T = 1.25$ msec, the second scenario for the VBR AR(2) video source had delay variances of 0.024 μsec^2, 0.022 μsec^2, and 0.019 μsec^2 for data, video, and audio (voice), respectively. This represents a 98% reduction for all 3 source types in the mean waiting time variance. Although the fixed DSC W/T parameter was not optimal for the AR(2) video source and thus allowed the source to retain more of its burstiness characteristic, the DSC W/T parameter still had a significant impact on the reduction of the delay variance.

For the case of cell percentiles with delays greater than 1 msec, the VBR AR(2) video source scenario had cell percentiles of 0.38902, 0.32879, and 0.2935 for data, video, and audio (voice), respectively, with the W/T parameters set at $W =$ 320 cells, $T = 50$ msec, . As the W/T parameters are lowered to $W = 8$ cells, $T =$ 1.25 msec, the second scenario for the VBR had cell percentiles with delays greater than 1 msec very near to 0 (zero) for all 3 source types with the values being 0.0016, 0.0013, and 0.0012 for data, video, and audio, respectively. This represents a reduction of 99.6% for all 3 source types. The reasons for the differences in the values when compared to Section 4.1.1 and 4.2.1 are the same as those for the delay and delay variance numbers described above.

4.2 Adaptive DSC Algorithm

For the adaptive DSC algorithm analysis, one of the network scenarios chosen for the non-adaptive case is used to test the performance of the adaptive DSC algorithm and is listed below:

- Variable Bit Rate AR(1) Video Source Model

The bursty data source is allowed to run with a non-adaptive DSC algorithm operating at a *W/T* parameter of 320 cells/50 msec. All other parameters for the adaptive DSC algorithm network scenario are the same as those stated for Section 4.1. This network scenario type was chosen to illustrate the performance improvement concept of the adaptive DSC algorithm.

4.2.1 Variable Bit Rate Video Source AR(1) Model Analysis

In this network scenario, the 45Mbps FBR video source on each 150Mbps ATM link in Section 4.1.1 is replaced with 12 VBR AR(1) video sources and operates the same as stated in Section 4.1.2.1.

At the higher *W/T* control, the delay average has a varying rate during the 5 to 15 second period of the simulation. The occurrence of both increasing and decreasing rates is due to the adaptive DSC attempting to compensate and lock onto the desired delay average. There is a slight decrease from the 20 to 55 second period. After 55 seconds, the delay levels out and remains in a steady state for the remainder of the 2 minute network simulation. The delays for all 3 data types are lower than the similar video source case in Section 4.1.2.1. In contrast to results of Section 4.1.2.1, the delay values for the bursty video sources track closer towards the FBR audio sources instead of the bursty data sources. Similar to the delay results, the percentile of cells greater than 1 msec delay start leveling off after the 55 second period of the simulation. The delay variance has a slight change in values during the 45 to 120 second period of the simulation. After 95 seconds into the simulation, the delay variance remains constant. Both of these measured parameters are lower than the case in Section 4.1.2.1.

With the adaptive *W/T* parameters, the adaptive DSC algorithm VBR video source AR(1) scenario had delays of 1016 μsec, 862 μsec, and 765 μsec for data, video, and audio (voice), respectively. When contrasted with the non-adaptive DSC algorithm case of Section 4.1.2.1, the delay average is reduced 22%, 36%, and 28% for data, video, and audio, respectively.

For the delay variance, the adaptive DSC algorithm VBR AR(1) video source scenario had delay variances of 1.478 μsec^2, 1.397 μsec^2, and 1.339 μsec^2 for data, video, and audio (voice), respectively. When contrasted with the non-adaptive DSC algorithm case of Section 4.1.2.1, the delay variance is reduced 27%, 30%, and 30% for data, video, and audio, respectively. For the case of cell percentiles with delays greater than 1 msec, the adaptive DSC algorithm VBR AR(1) video source scenario had cell percentiles of 0.3566, 0.2954, and 0.2667 for data, video, and audio (voice), respectively. These values represent a 20%, 37%, and 27% reduction for data, video, and audio, respectively.

4.3 Computer Simulation Results

The computer simulation results of the various network scenarios are provided in the following pages. There are several sets of graphs for the two minute network analysis and each set includes 3 graphs consisting of:

1. Delay Average
2. Delay Variance
3. Percentile of Cells with Delays greater than 1 milliseconds

Data from the following network scenarios are provided:

1. Fixed Data DSC ($W/T = 320/50$)
2. Fixed Data DSC ($W/T = 8/1.25$)
3. Fixed DSC (Data $W/T = 320/50$, AR(1) Video $W/T = 390/50$)
4. Fixed DSC (Data $W/T = 8/1.25$, AR(1) Video $W/T = 10/1.25$)
5. Fixed DSC (Data $W/T = 320/50$, AR(2) Video $W/T = 390/50$)
6. Fixed DSC (Data $W/T = 8/1.25$, AR(2) Video $W/T = 10/1.25$)
7. Fixed DSC Data $W/T = 320/50$ and Adaptive AR(1) Video W/T)

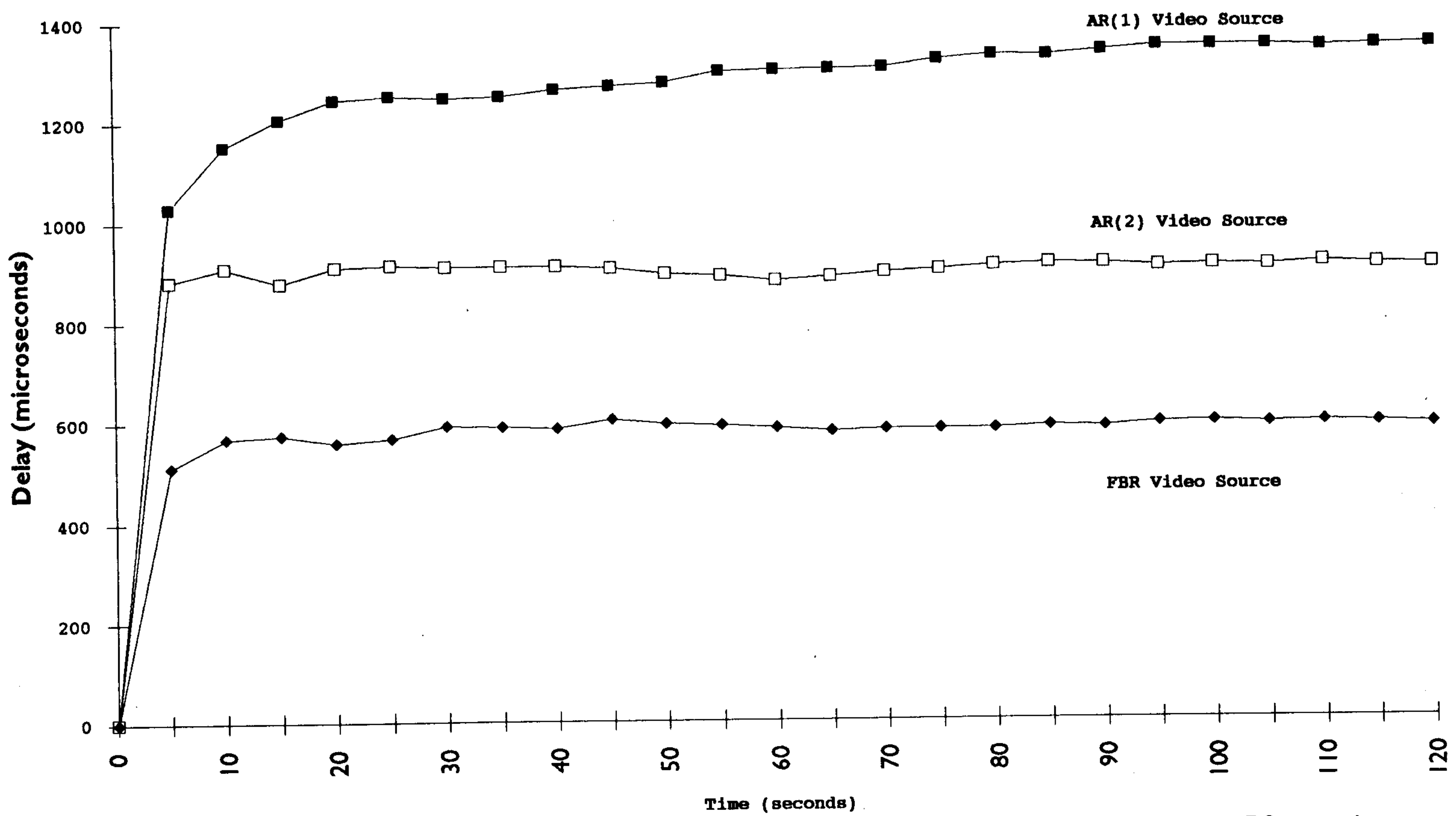

Figure 4.1: Video Source Average Delay (Non-adaptive DSC, T=50msec)

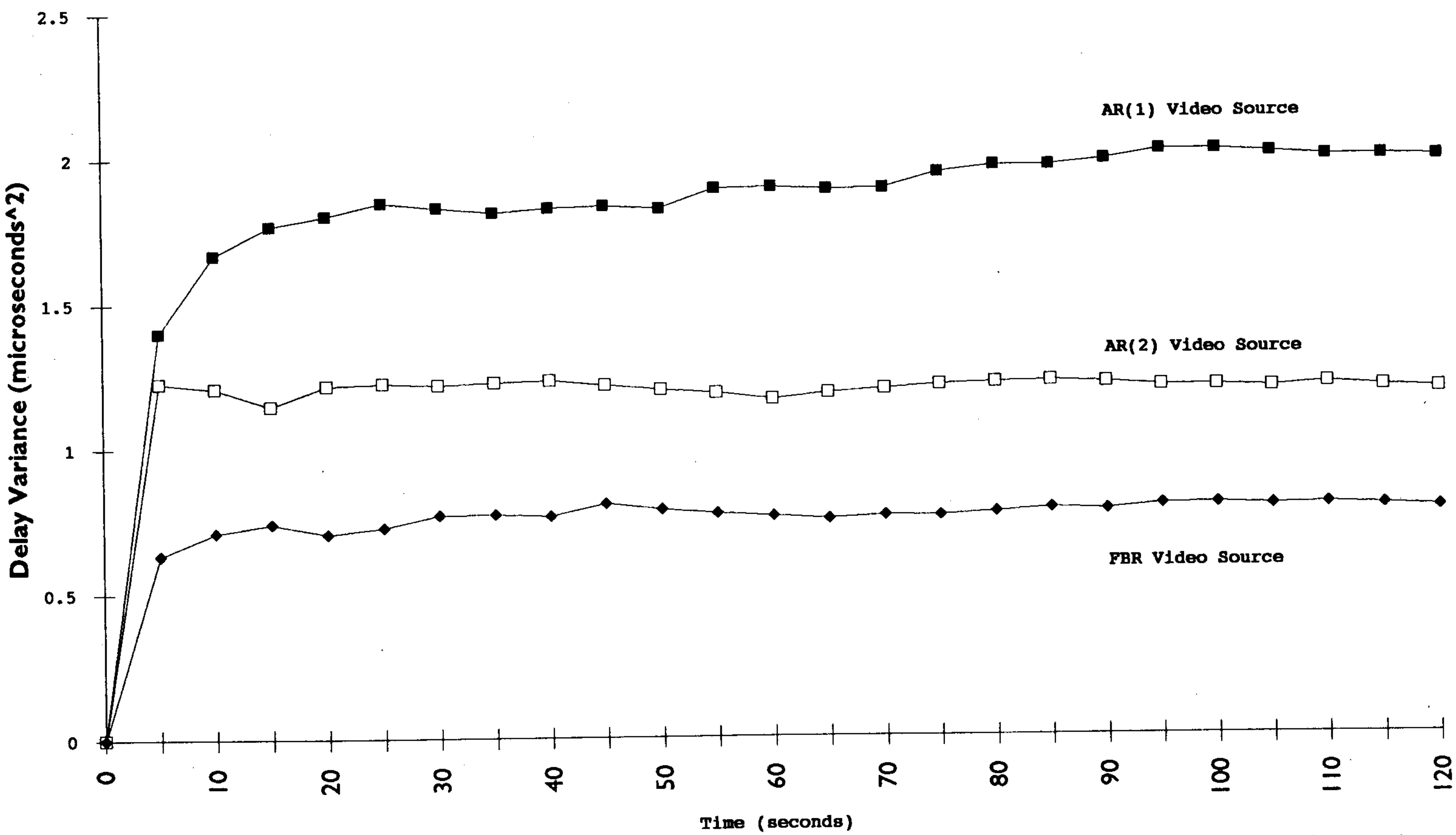

Figure 4.2: Video Source Average Delay Variance (Non-adaptive DSC, T=50msec)

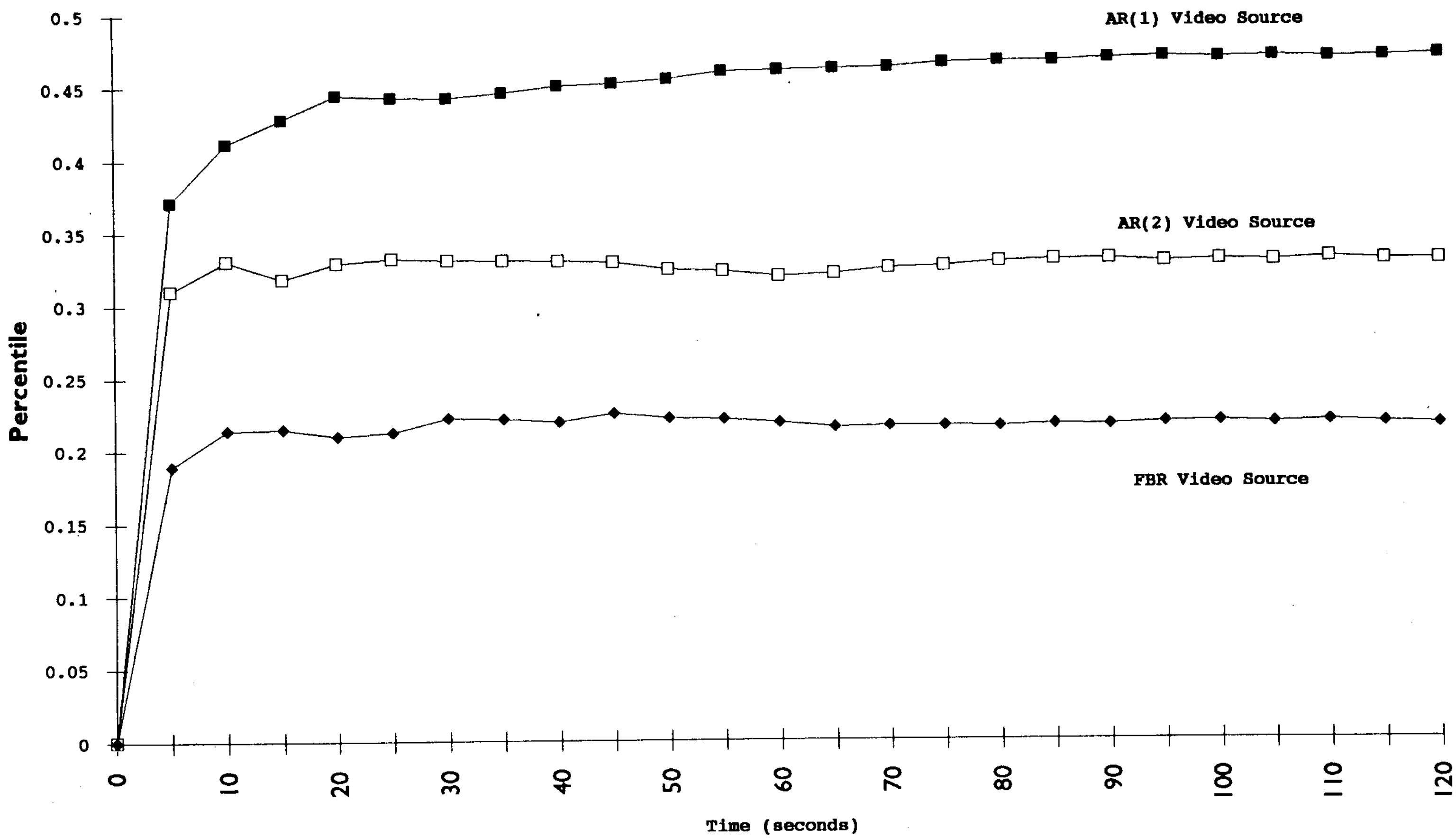

Figure 4.3: Video Source's Percentile of Cells>1 msec (Non-adaptive DSC, T=50ms)

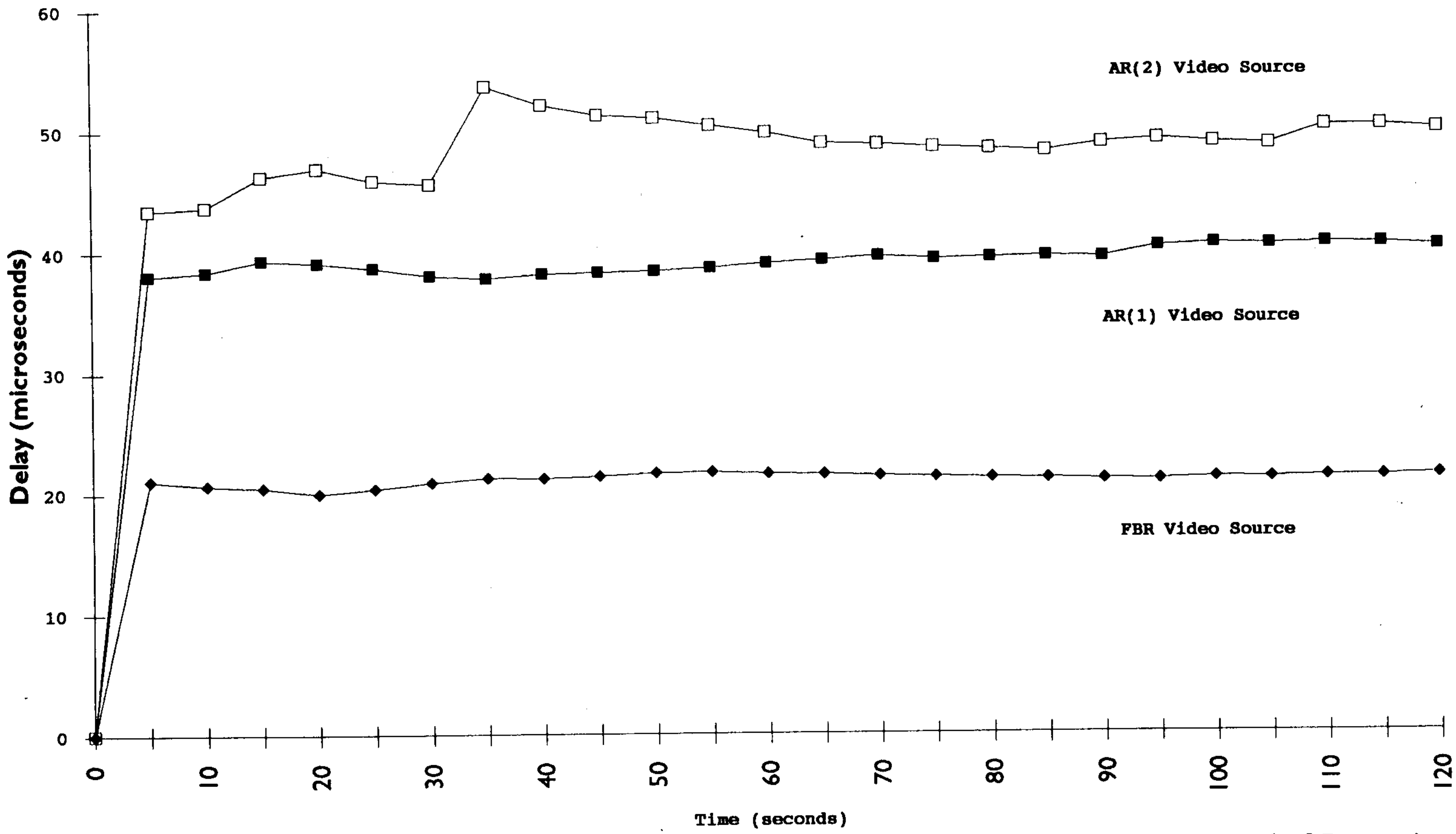

Figure 4.4: Video Source Average Delay (Non-adaptive DSC, T=1.25msec)

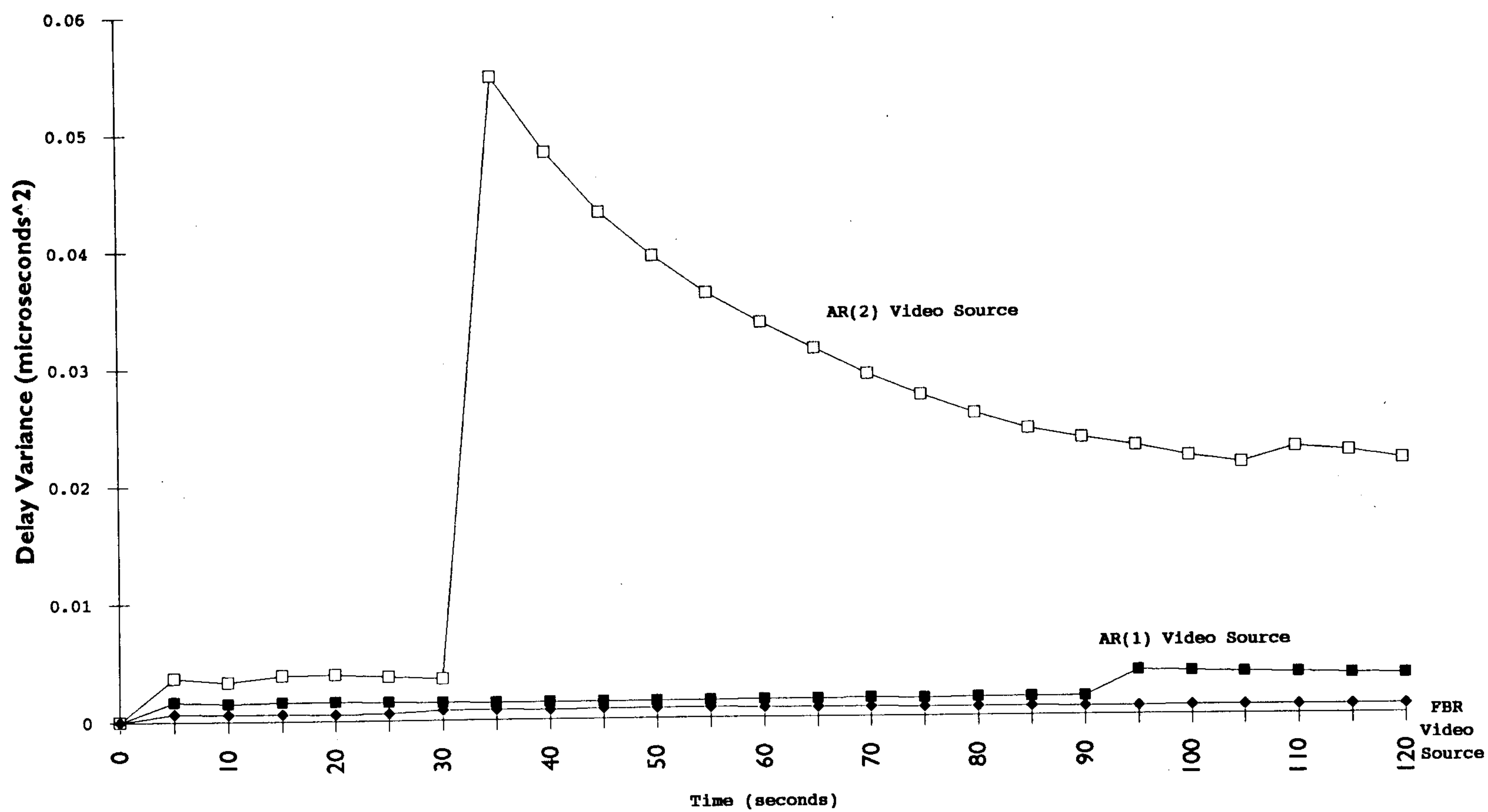

Figure 4.5: Video Source Average Delay Variance (Non-adaptive DSC, T=1.25ms)

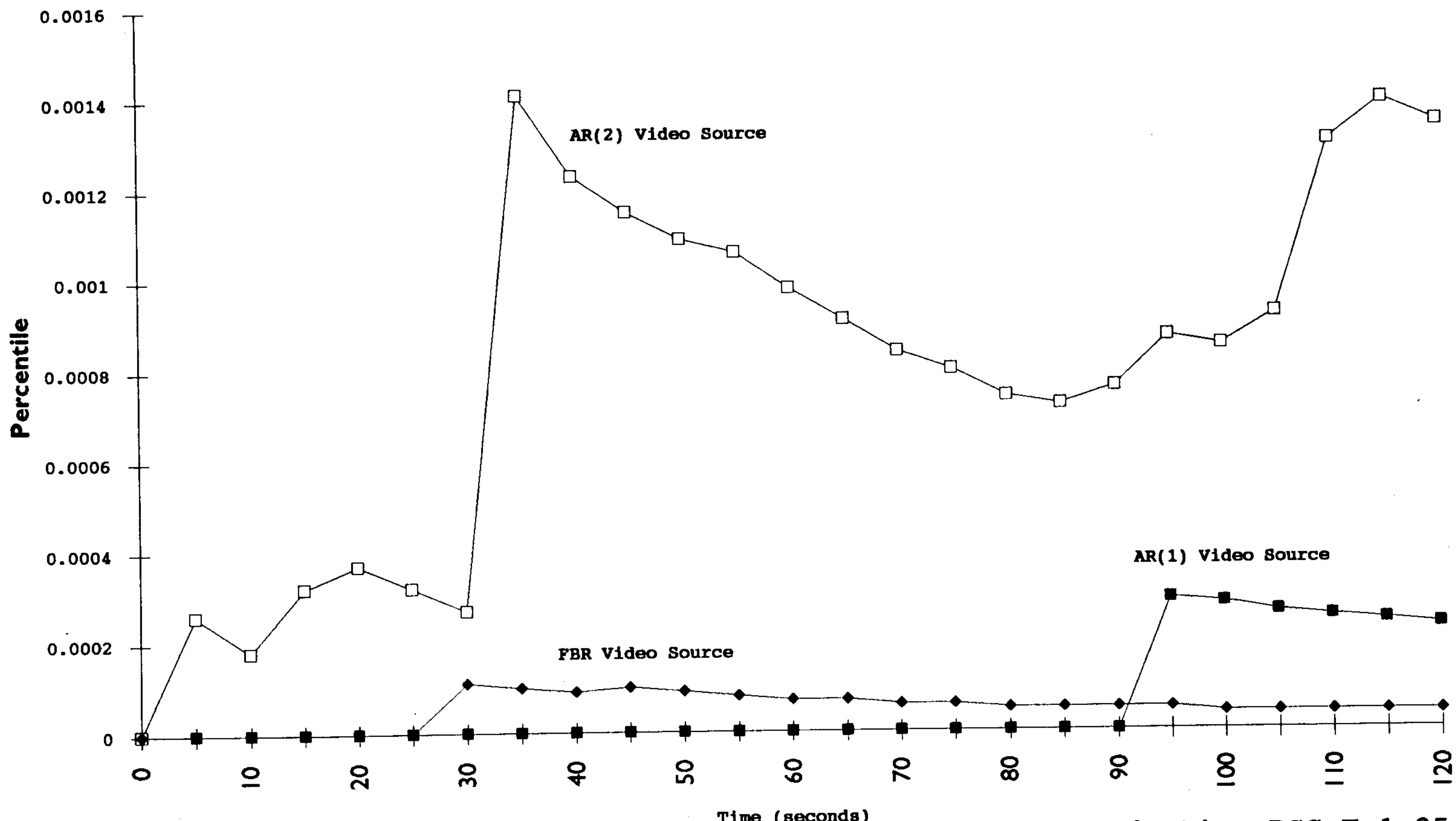

Figure 4.6: Video Source's Percentile of Cells>1 msec (Non-adaptive DSC,T=1.25ms)

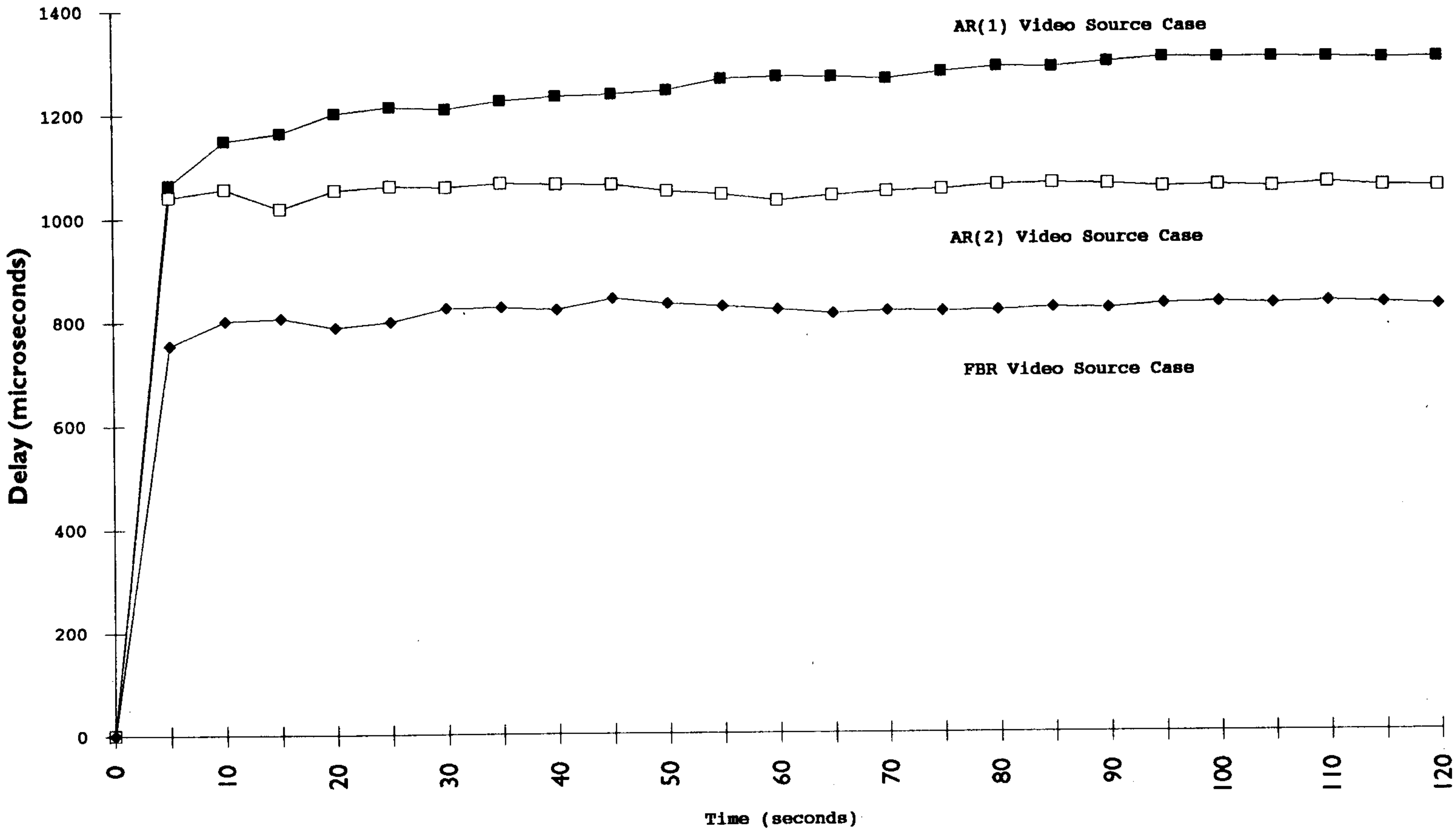

Figure 4.7: Data Source Average Delay (Non-adaptive DSC, T=50 msec)

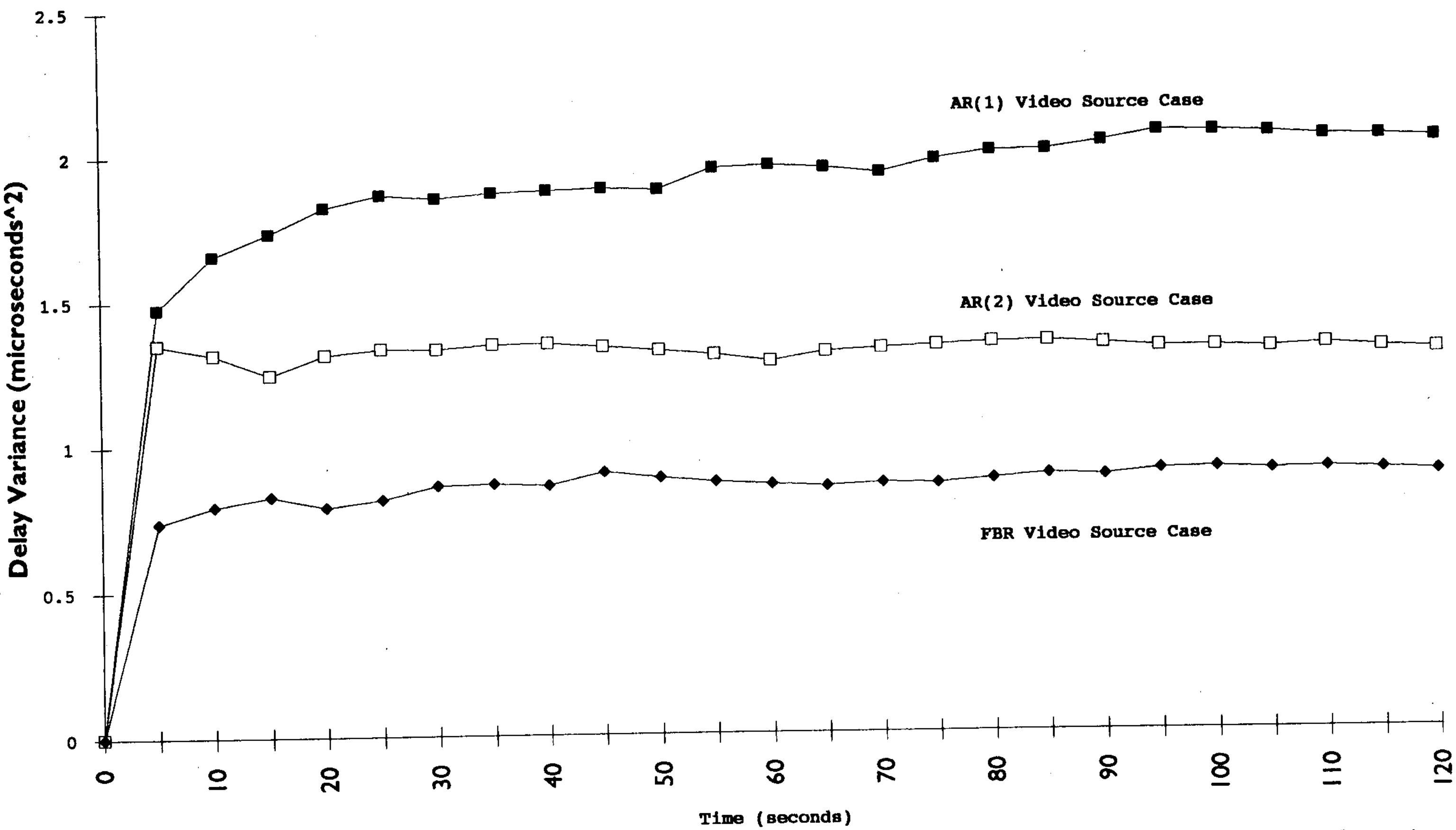

Figure 4.8: Data Source Average Delay Variance (Non-adaptive DSC, T=50 ms)

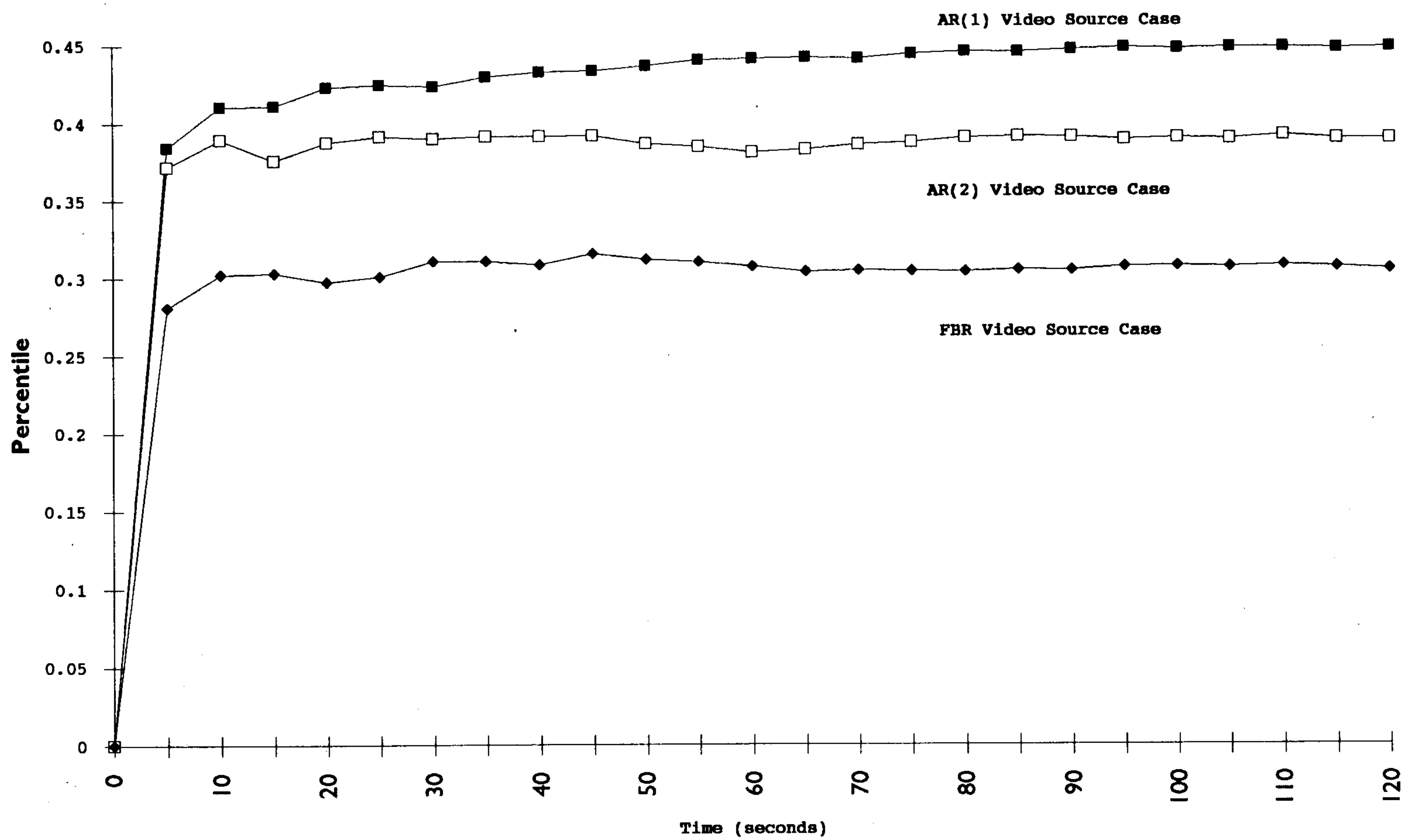

Figure 4.9: Data Source's Percentile of Cells>1 msec (Non-adaptive DSC,T=50 ms)

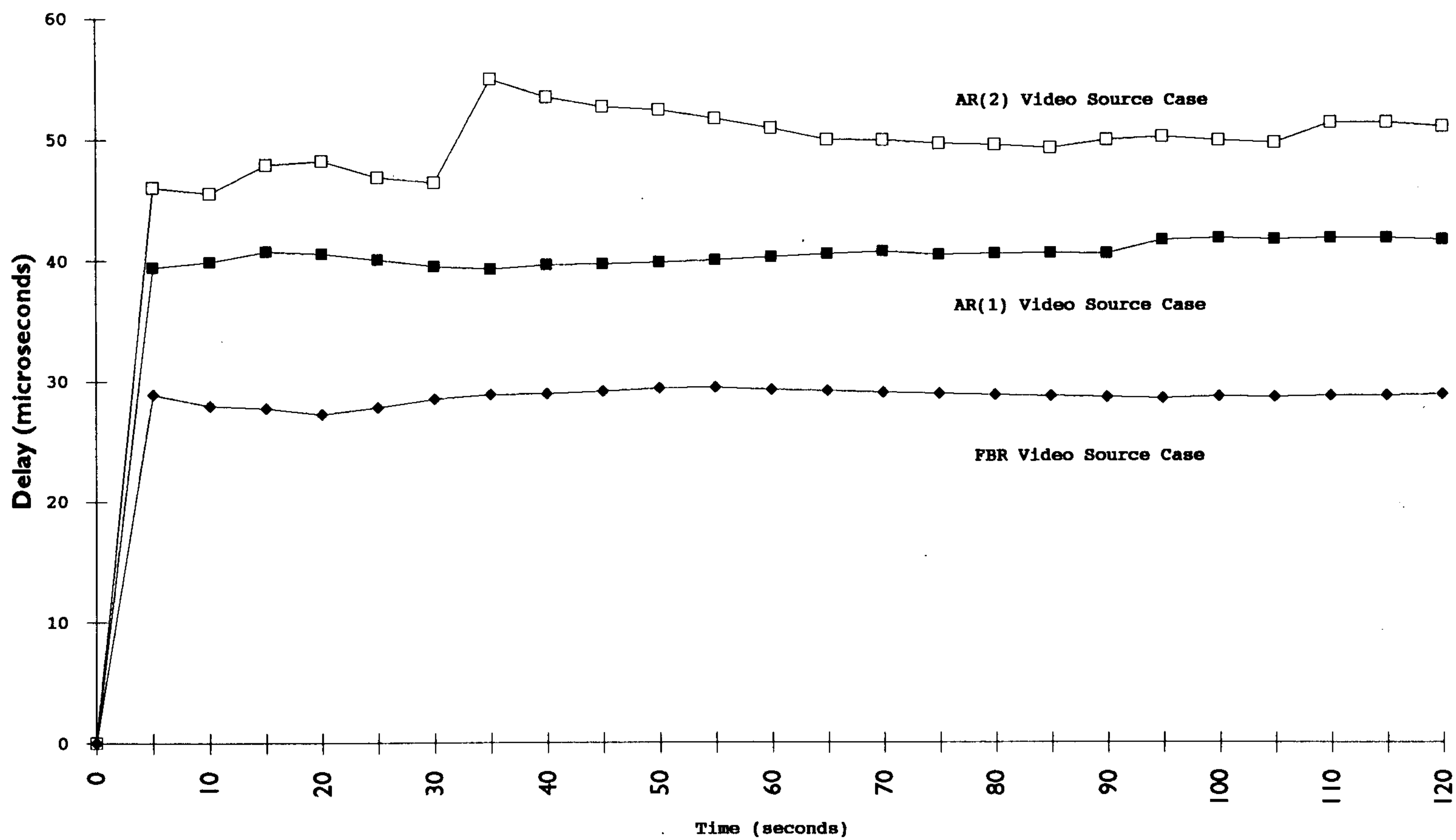

Figure 4.10: Data Source Average Delay (Non-adaptive DSC, T=1.25msec)

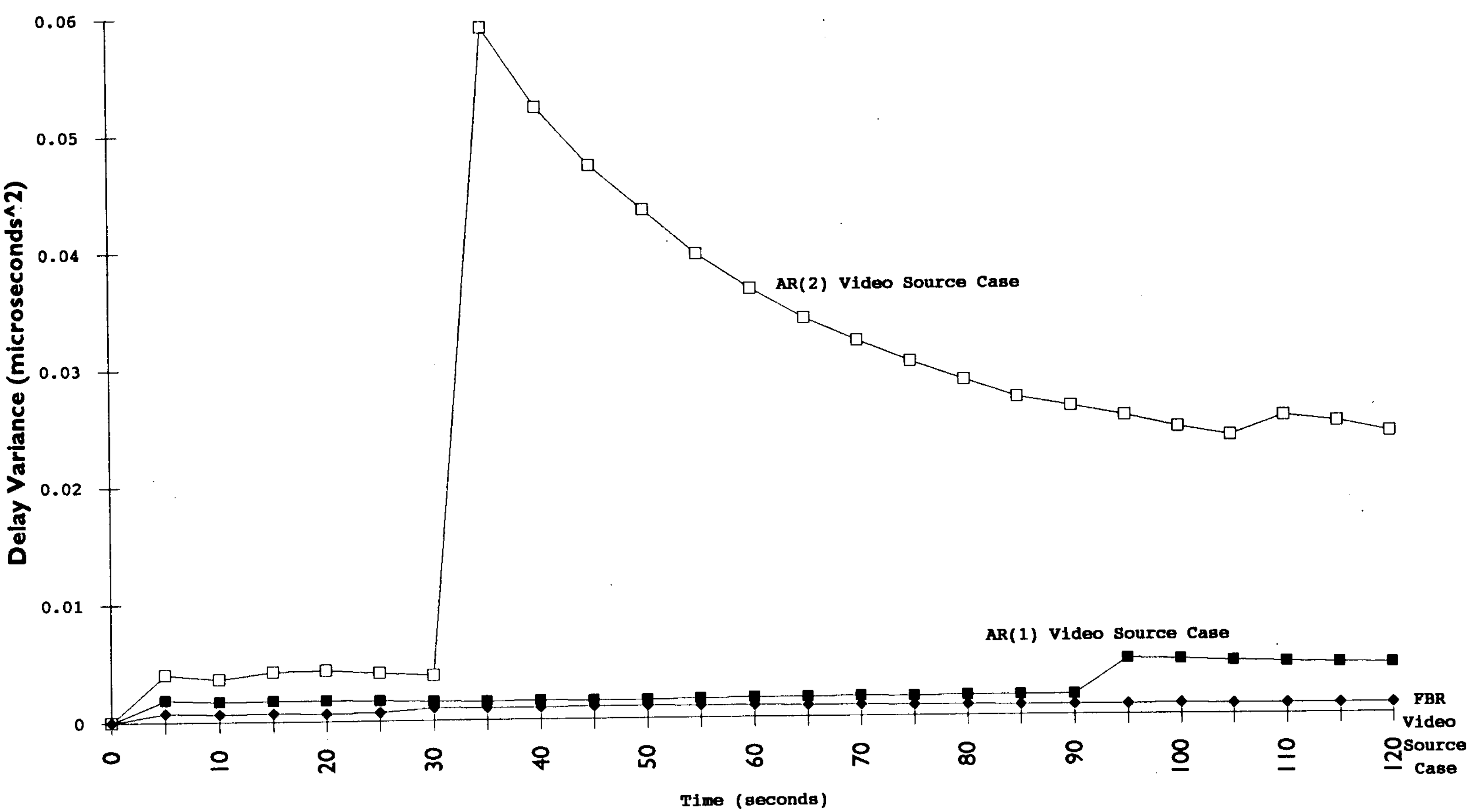

Figure 4.11: Data Source Average Delay Variance (Non-adaptive DSC, T=1.25ms)

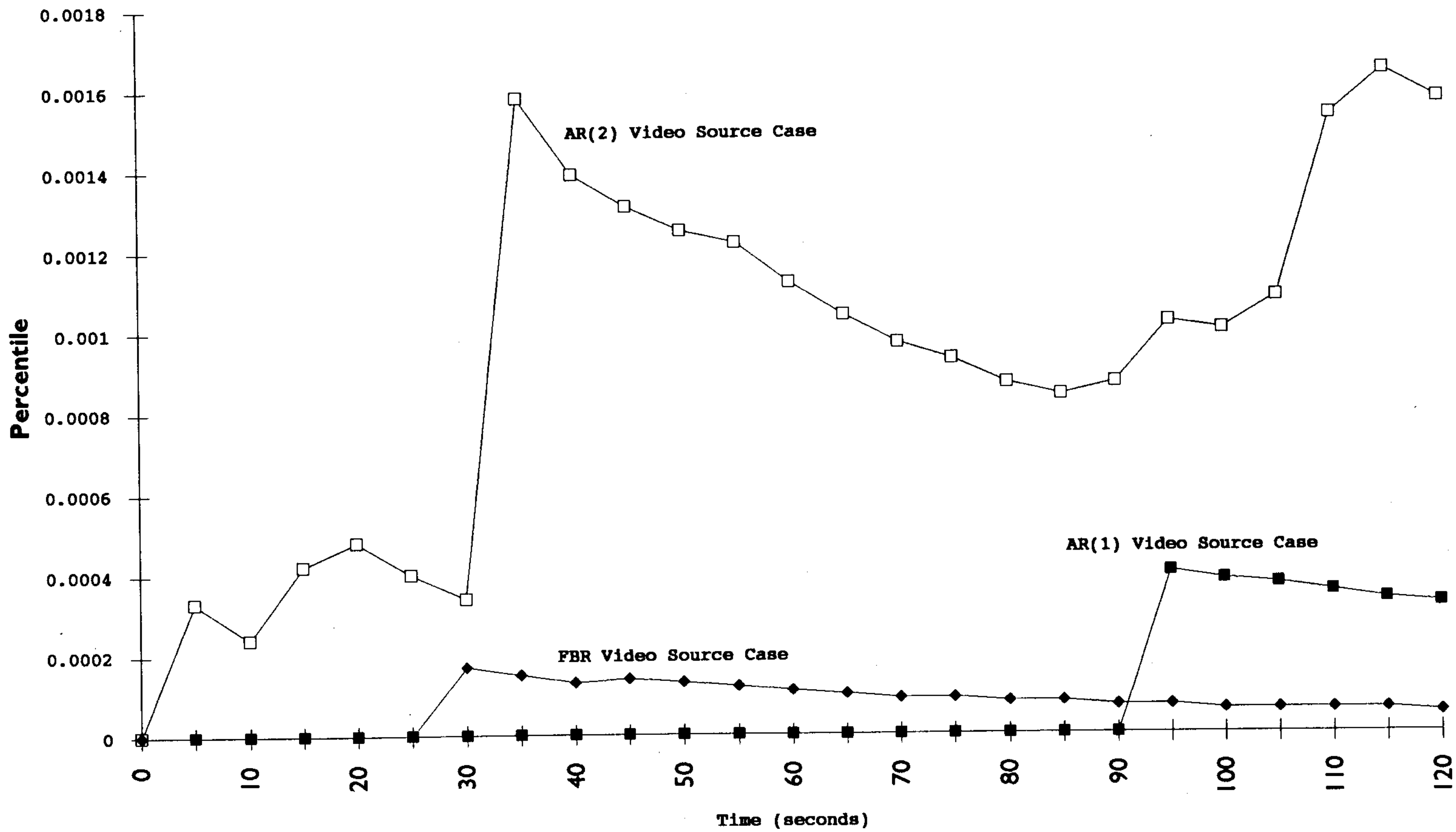

Figure 4.12: Data Source's Percentile of Cells>1 msec(Non-adaptive DSC,T=1.25ms)

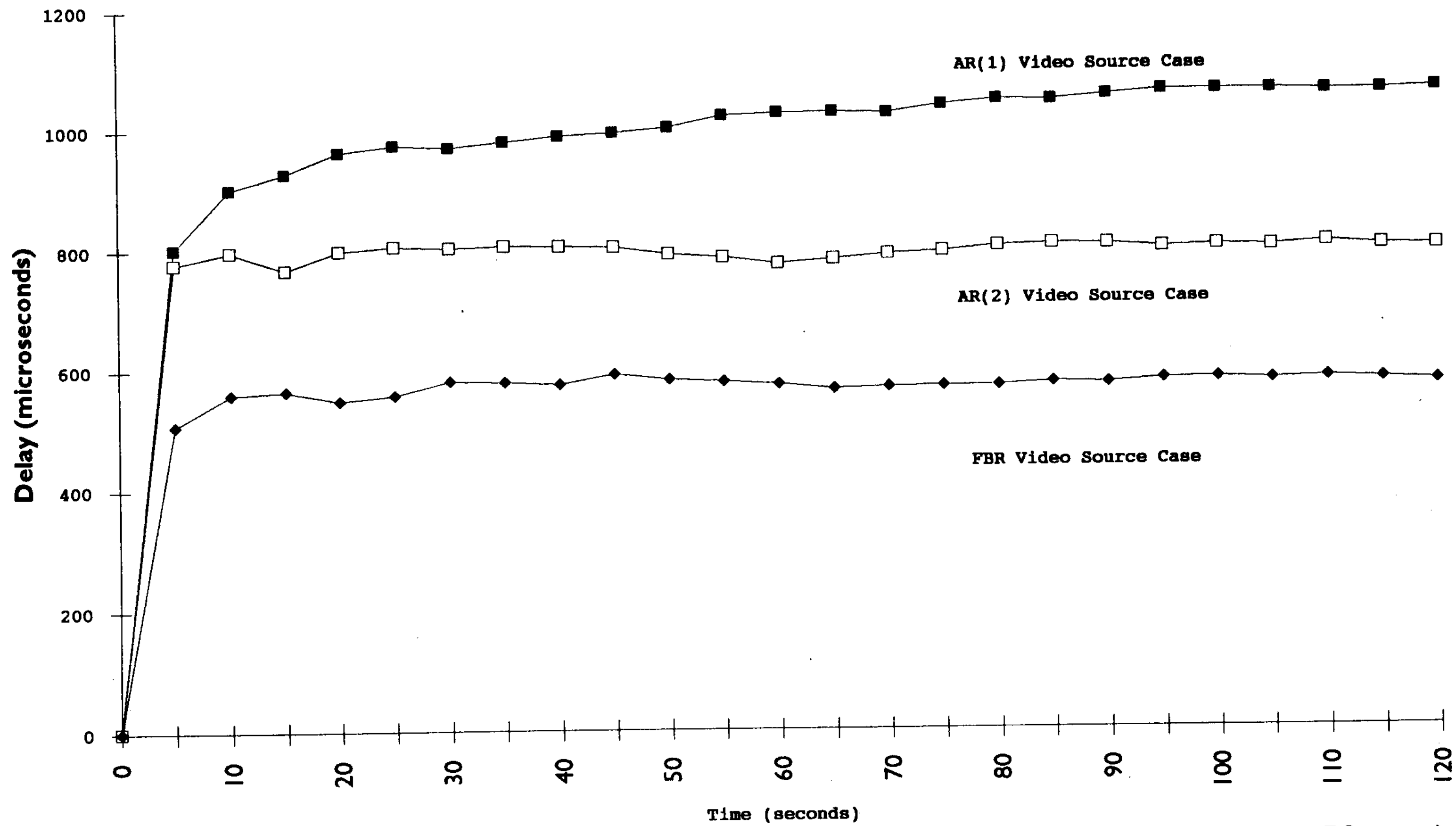

Figure 4.13: Audio Source Average Delay (Non-adaptive DSC, T= 50msec)

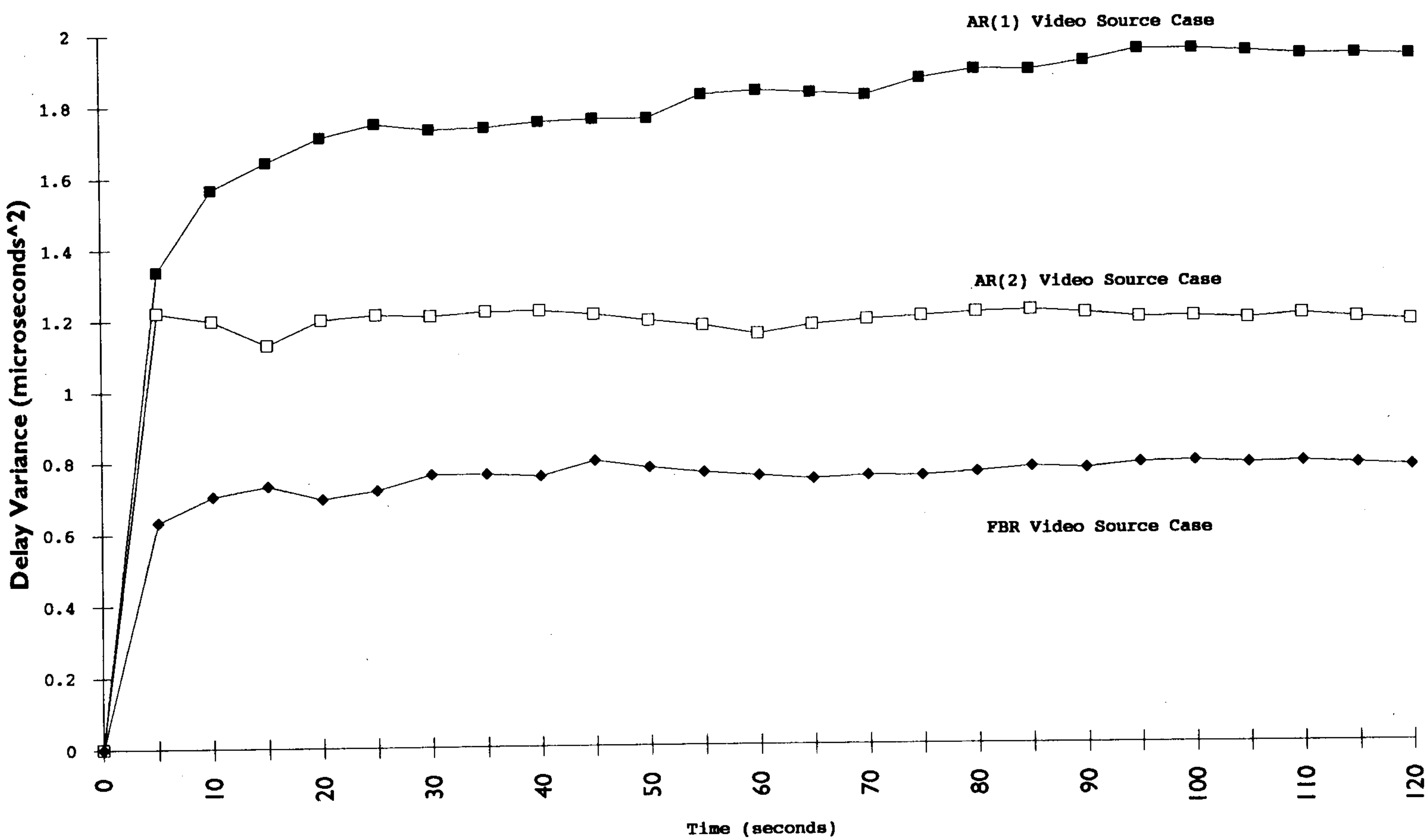

Figure 4.14: Audio Source Average Delay Variance (Non-adaptive DSC, T= 50ms)

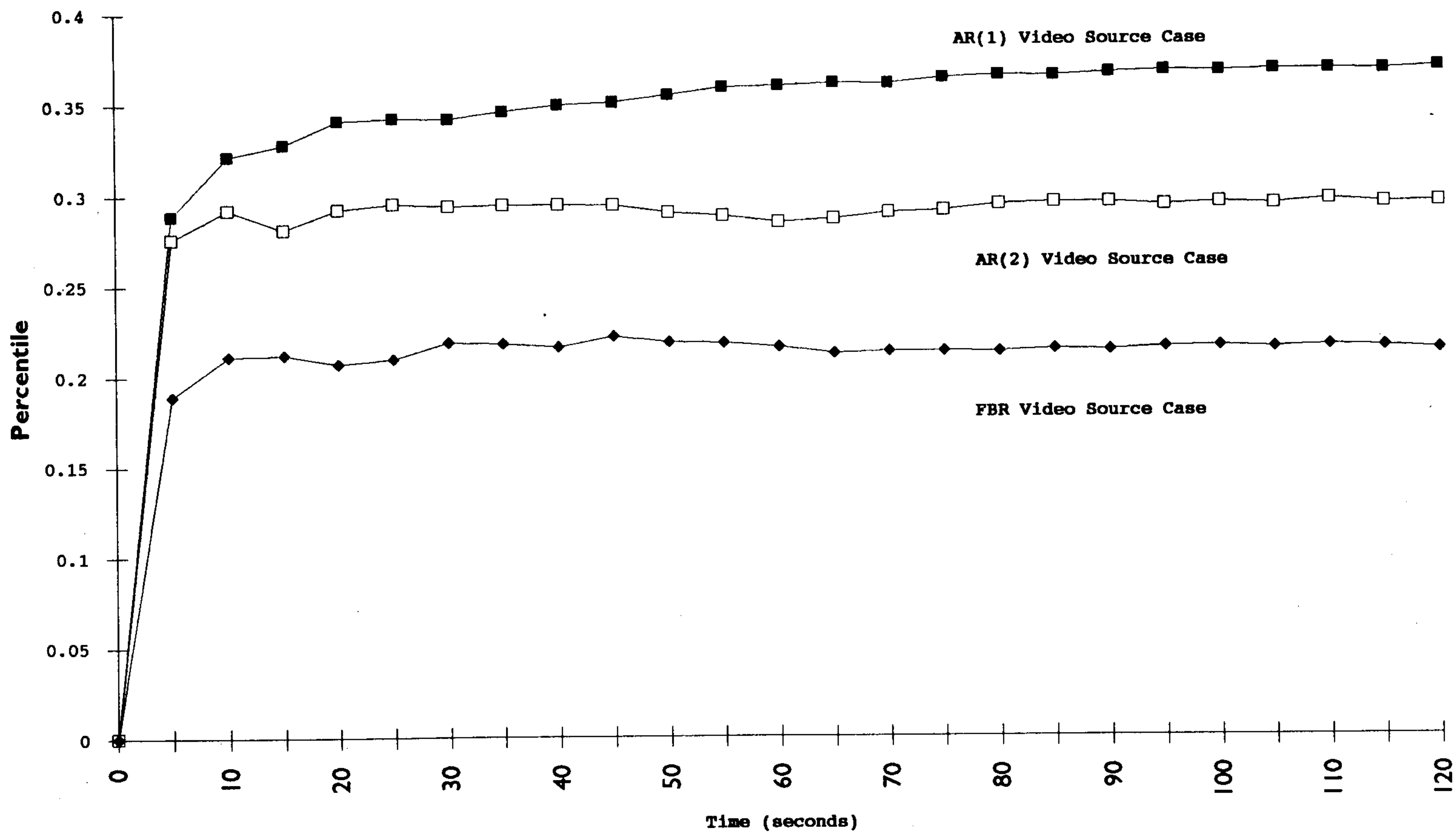

Figure 4.15: Audio Source's Percentile of Cells>1 msec (Non-adaptive DSC,T=50ms)

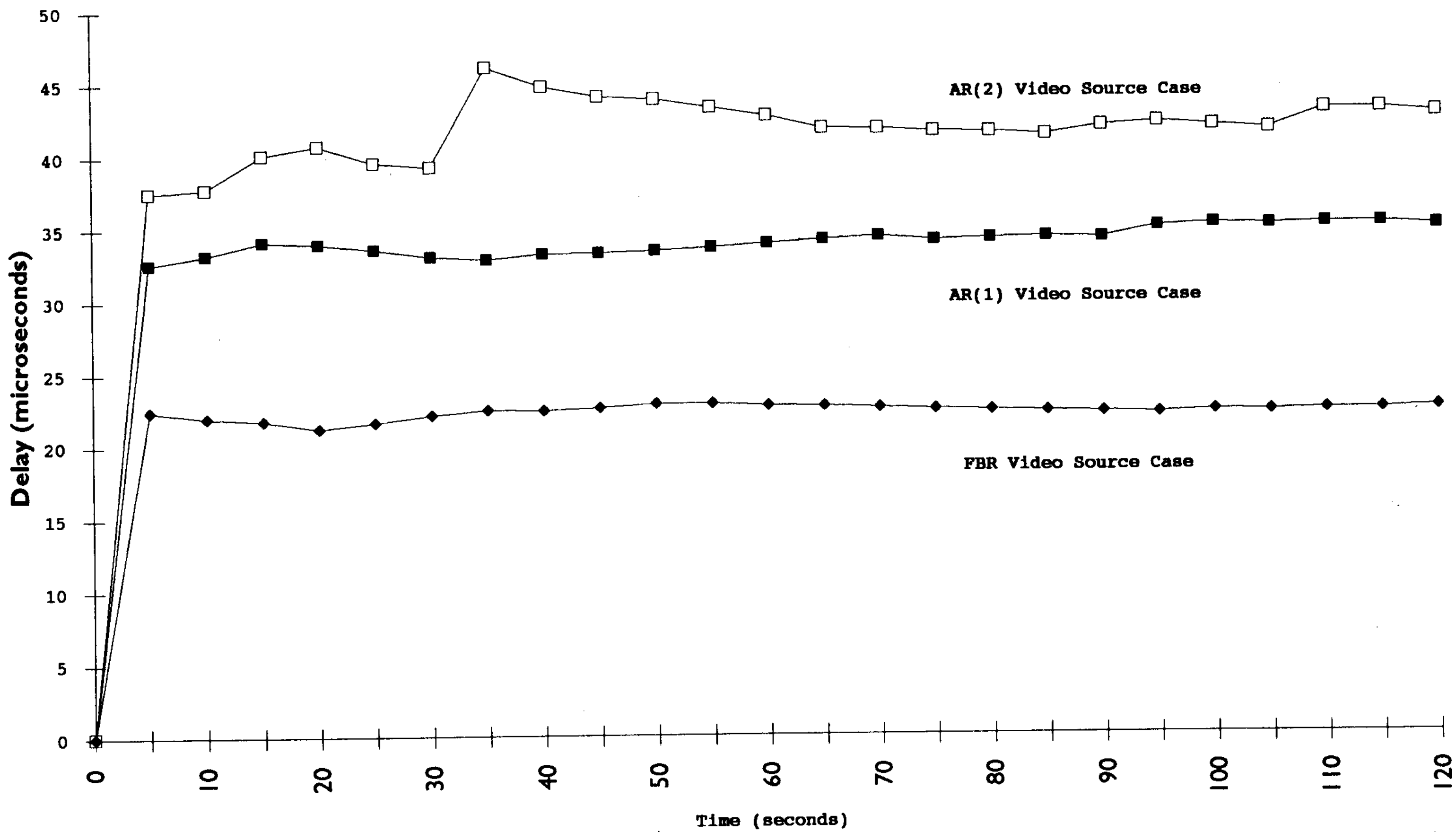

Figure 4.16: Audio Source Average Delay (Non-adaptive DSC, T=1.25msec)

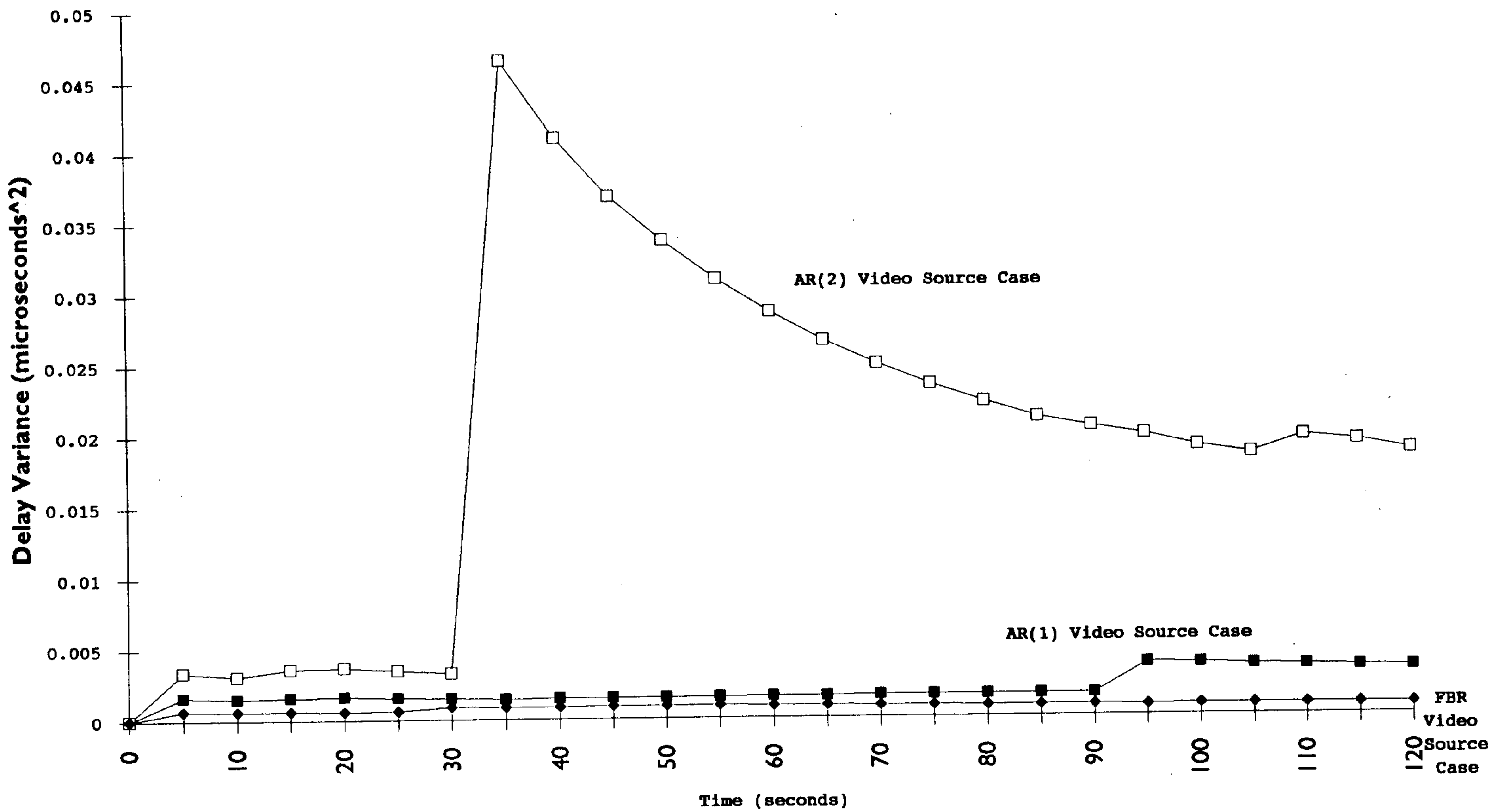

Figure 4.17: Audio Source Average Delay Variance (Non-adaptive DSC, T=1.25ms)

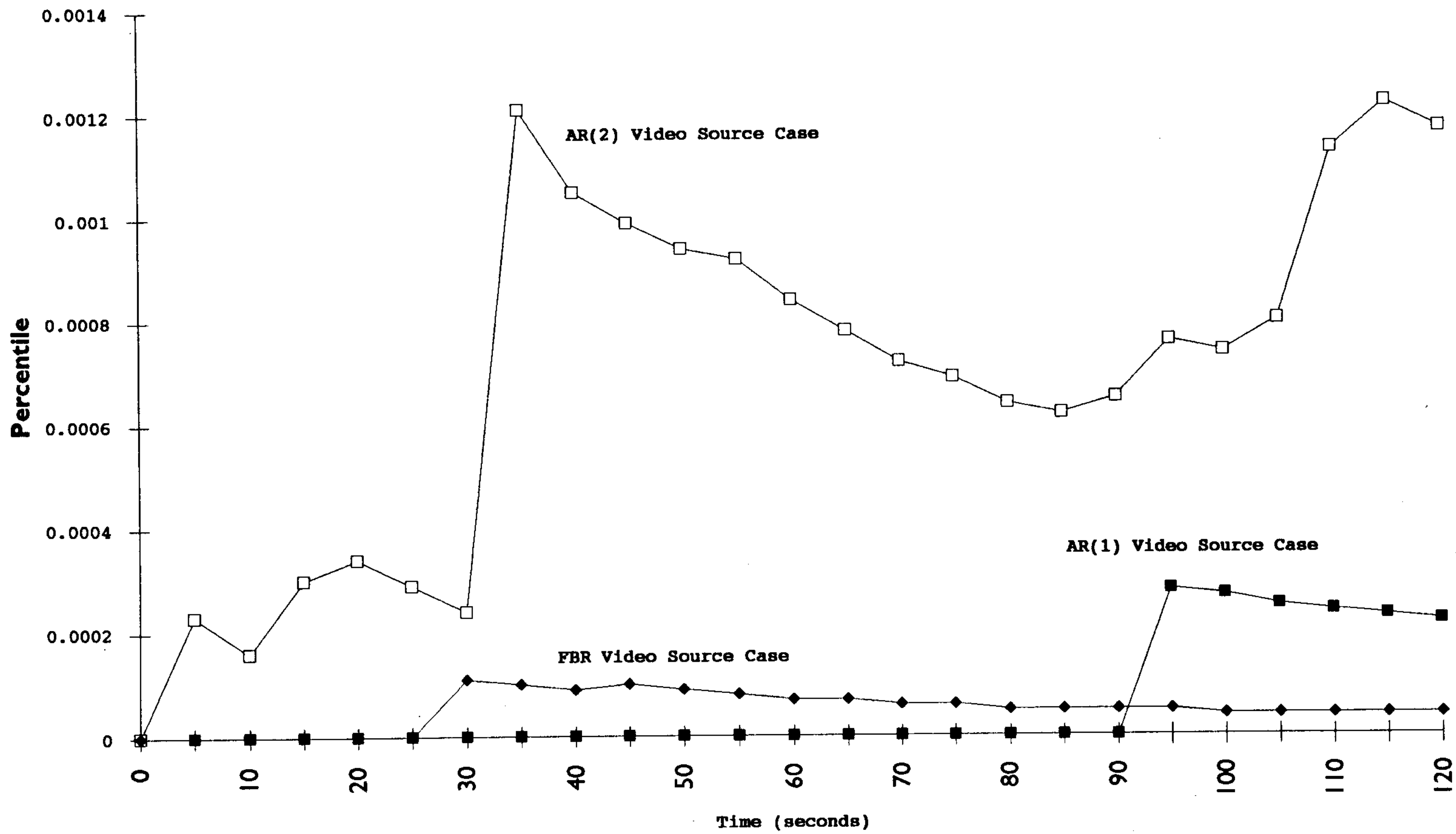

Figure 4.18: Audio Source's Percentile of Cells>1 msec(Non-adaptive DSC,T=1.25ms)

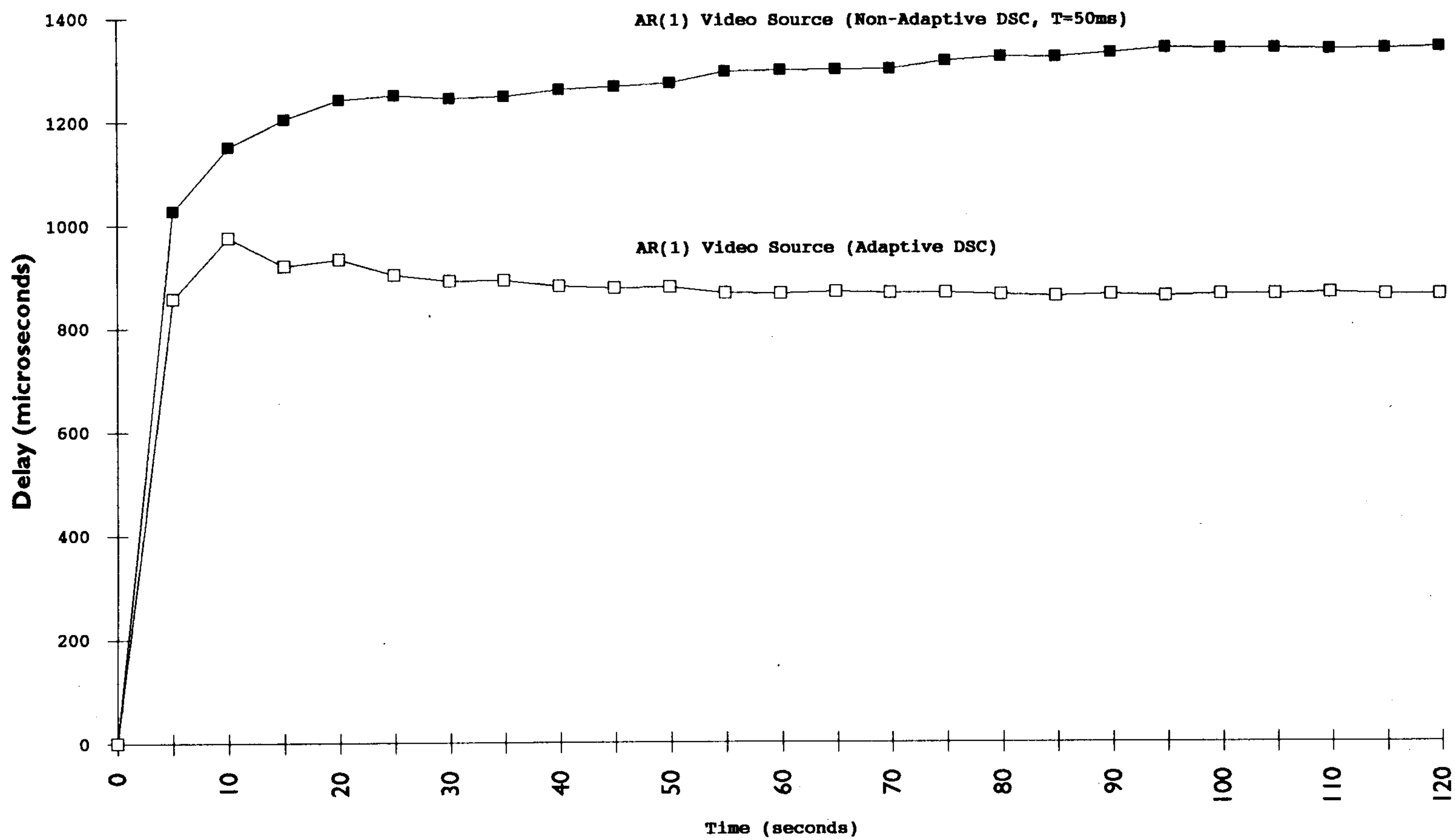

Figure 4.19: Video Source Average Delay (Non-adaptive vs. Adaptive DSC)

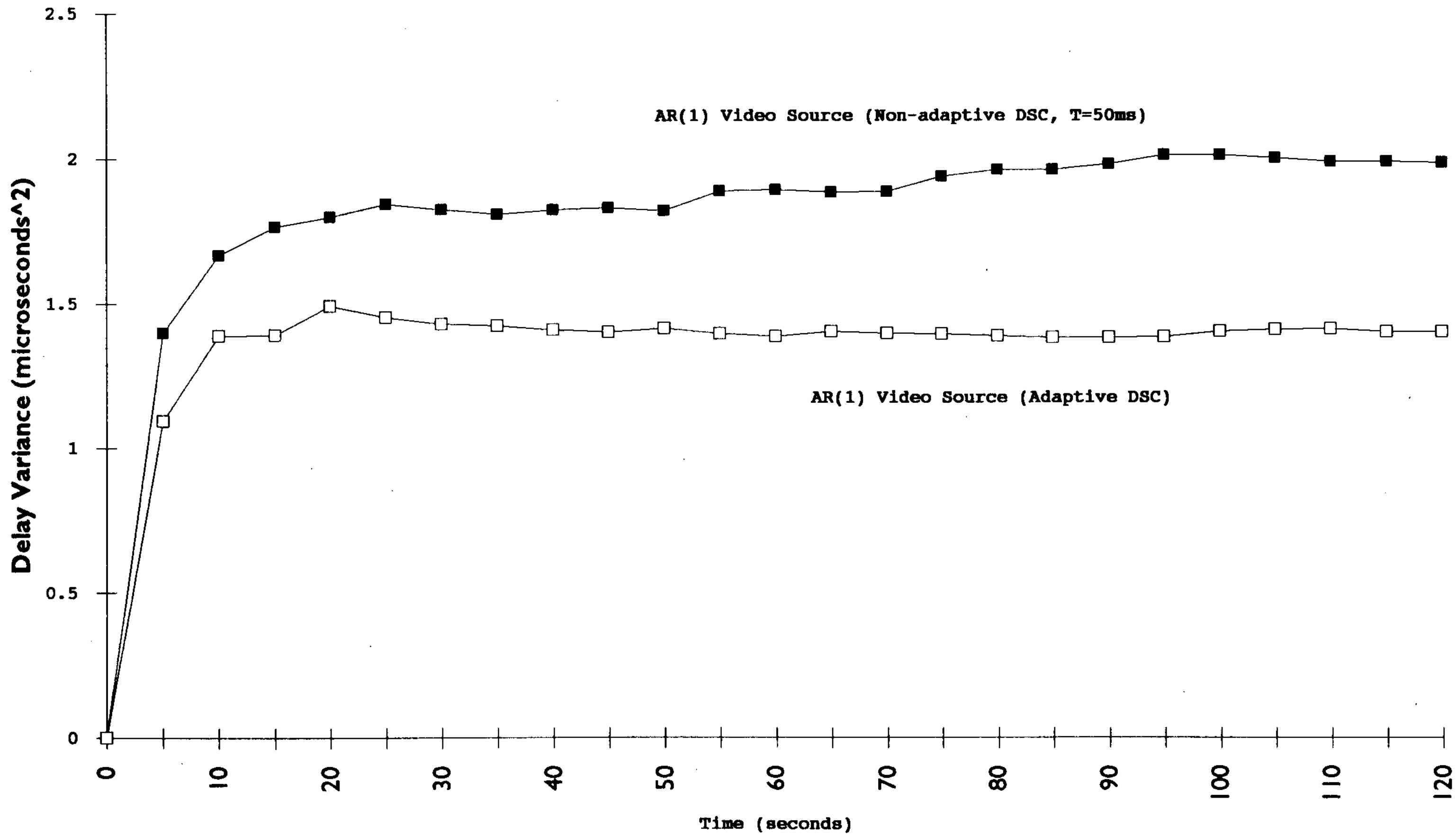

Figure 4.20: Video Source Delay Variance(Non-adaptive DSC vs. Adaptive DSC;Avg.)

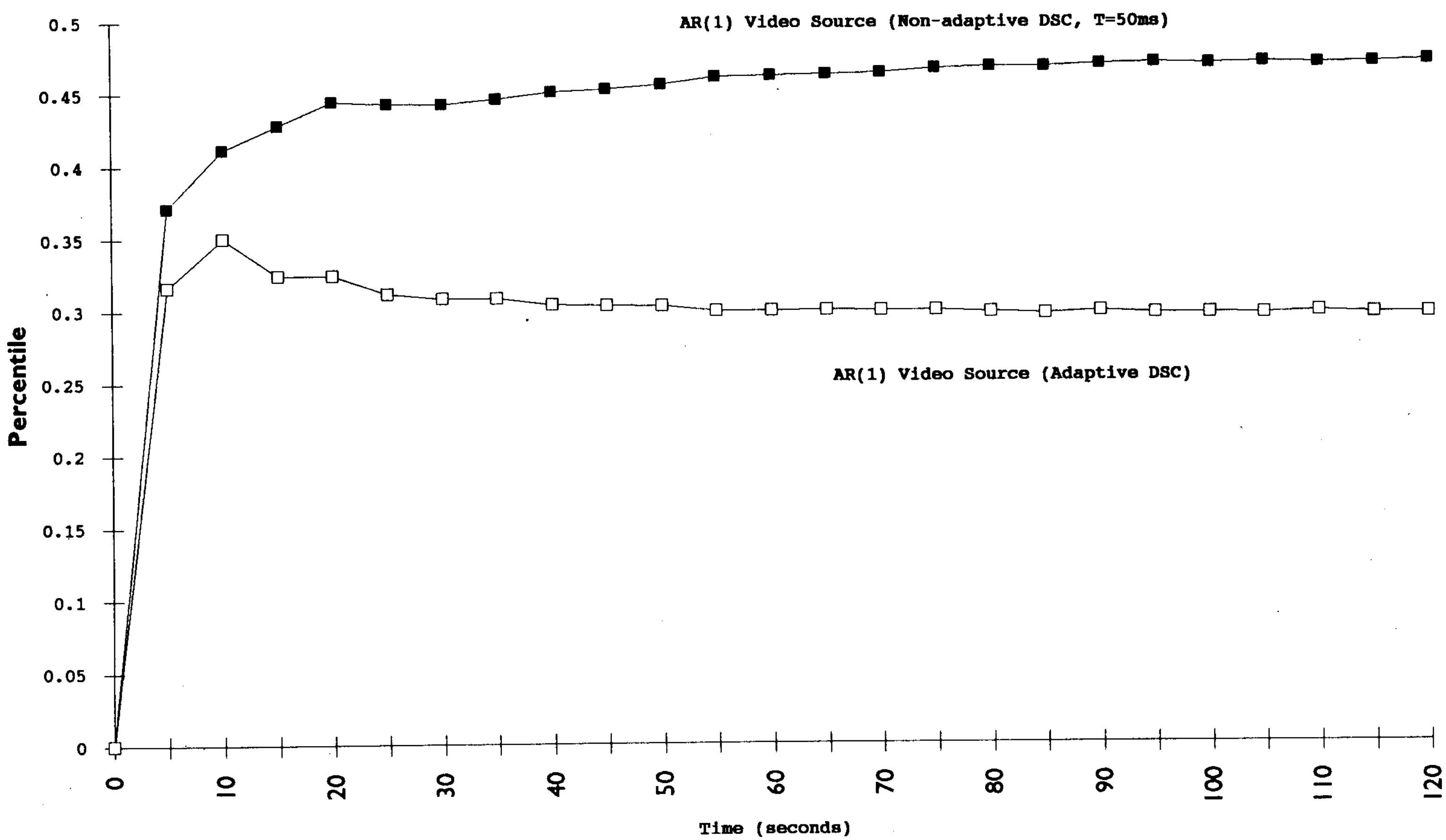

Figure 4.21:Video's Percentile of Cells>1 msec(Non-adaptive DSC vs. Adaptive DSC)

CHAPTER 5

CONCLUSIONS

This thesis investigated several video source models, selected two models, and then performed a computer simulation of an ATM network with multimedia data sources: video, voice, and data. The available video source models all focus on video teleconferencing applications. This application is currently a more common application for video over networks. Other video applications exist and if additional information existed on the particulars of these video sources, a couple of video source models (from [8] and [10]) are capable of incorporating some of the basic characteristics of more advanced video applications. Thus, the current constraint that limits network traffic studies to video teleconferencing examples can be removed to include more advanced video applications. However, this constraint can only be lifted if the research community, in a future study, defines a set of video time sequences for these advanced video applications. This will allow for more thorough traffic studies involving full-motion video services.

The two video source models considered here consisted of an autoregressive AR(1) and AR(2) video model source. The conversion of the fixed bit rate video source to a variable bit rate video source (autoregressive based model) degraded network performance. Between the two sources using a non-adaptive DSC protocol, the AR(1) video source provided higher network performance numbers (lower numbers are preferable) at the high DSC *W/T* parameter values and would be recommended for worst case analysis with the adaptive DSC protocol. The AR(2) video source model provided higher network performance numbers at the same low DSC *W/T* parameter settings. This difference was attributed to the fixed DSC *W/T* parameters being set for the AR(1) video statistics, which resulted in a lower percentage reduction for the AR(2) performance as the DSC control parameters were changed from the high (390/50ms) to the low *W/T* values (10/1.25ms). On the basis of the simulation results, the AR(2) would not be recommended over the AR(1) unless the simulation machine had sufficient processing power. This is due to the AR(2) requiring additional video sources per link to maintain the same level of link payload rates. The additional video sources slow down the simulation in terms of managing the additional video sources. Both video models provide challenging sequences of video cells when allowed to run for a significant amount of time such as the two minute computer simulation. These models provide intense demands on network management implementations and require robust network management control schemes for effective network performance. From the simulation results, the AR(1) video model will suffice for network analysis focusing on cell delay and delay variance.

The concept of DSC's rate and policing control to real-time sources is shown to be feasible in an adaptive form that adjusts to the continuously changing instantaneous source rate. The proposed adaptive DSC protocol addresses many of the major technical issues related to supporting high speed network video services including delay, jitter, and congestion (overflow). The baseline adaptive DSC protocol was defined at a network system level to fine tune the traffic smoothing at the source based on feedback from the source itself and from the network. Comparison of the DSC algorithms indicates that the adaptive DSC algorithm approach is the more desirable approach due to its performance improvement and its ability to adjust control parameters to the specific distribution of the source. Without the adaptive capability, the non-adaptive DSC algorithm would require fixed initial settings at peak rate settings for a *generic* video source to ensure that the cells do not encounter excessive discarding at the source due to a mismatched DSC setting.

From the results of the fixed and non-adaptive DSC protocols, the fixed DSC protocol is recommended primarily for non real-time data sources when considering the following:

1. The lower the DSC *W/T* value, the better the overall network performance
2. Competing DSC *W/T* values may adversely affect another data type. This adverse effect may arise when each data type passes through the same network node queue instead of having separate virtual queues for each data type or call. The competing DSC values can create a situation where the queue's percentage of the buffer currently being utilized for storing data is large and thus the queuing delay is relatively long. Another data type having less stringent QOS parameters for an adaptive DSC control can cause this occurrence. A common, fixed DSC *W/T* setting provides an equitable treatment to all non real-time data sources.

The adaptive DSC protocol is recommended for real-time sources and a summary of the recommendations follows:

1. Time constant should be fixed low, or only allowed to vary slightly:
   - Higher to obtain better resolution link utilization rate

- Lower to assist in a potential network queue overflow situation along a call's path

2. The queue overflow protection should be based on percentage of buffer size and input link utilization rate

3. Adaptive *W* parameter based on:

   - A monitored average *W* for the call with at least one relatively short term time constant
   - Source queue delay
   - Potential buffer overflow

4. Adaptive DSC protocol should be used for the case when the actual source statistics are not known prior to network transmission

5. Separate virtual queues are made for the individual calls to prevent other calls *W*/*T* settings from affecting significantly the call under consideration

6. DSC protocol design provides enough flexibility such that consideration can be given to integrating an existing network management scheme with the DSC algorithm, in a future study. Integration of other schemes with the DSC protocol is applicable when addressing situations that may arise for potential source queue overflow. These situations will typically occur during the time periods that contain a long sequence of video cells per frame, with the cells per frame rate (or bits per frame) being near the source's peak bit rate. The video models provide a significant challenge for any network management scheme. Thus, for the case of managing video sources, more than one type of network management tool may

be required to maintain a high level of QOS for the source while still maintaining a certain QOS for other sources using the network.

The fixed DSC algorithm can be made to work at its given optimum by having the lowest *W/T* value based on the link clock rate and constrained by either a W_{min} =1 or a T_{min}. The adaptive DSC algorithm can be made to work at its optimum performance by fixing *T* and allowing the window of cells to follow closely the average number of cells per *T*. A slight padding should be included in *W* to be on the plus side of the measured average to ensure that the source queue does not grow continuously over time. The fixed DSC algorithm can be made to out perform the adaptive DSC algorithm if the statistics are know prior to the call setup. Otherwise, the adaptive DSC will typically be superior for cases where the statistics are not known prior to the call.

The adaptive DSC protocol provides traffic smoothing in an attempt to avoid network congestion and prevent the discarding of cells to achieve its primary set of goals. Also, based on the source and network feedback, the adaptive DSC protocol control parameters can change in an attempt to meet the stated QOS requirements. In addition, while still maintaining the same effective average transmission rate, the DSC *W/T* parameter can be used to help remove congestion within the network by lowering the *W* and *T* parameters by equal amounts. The adaptive DSC protocol, which is defined to work either at the source or access node along with working closely with the periodic feedback from the network and destination nodes, provides a robust network management scheme. This scheme can be integrated directly into a network or, due to its modularity, the adaptive DSC protocol can be made to work in conjunction with existing network management schemes to utilize the strengths of all these algorithms.

APPENDIX A

COMPUTER SIMULATION SUMMARY

The ATM computer simulation consists of the following:

- 4 Nodes per hop
- 3 Hops (Access, Intermediate, Destination Nodes)
- One 150 Mbps link connected to each access node
- Each node's output can be sent to any of the 4 nodes in the next hop depending upon initial call setup
- ATM Cell size of 69 Bytes
  - 64 Byte ATM Payload
  - 5 Byte ATM Header
- Number and Type of Sources
  - 32 Host Data Types per access input link for a total of 128 sources
    - Hyperexponential Distribution with mean of 20KB per data file and squared coefficient of variation equal to 25. The approximate mixture consisted of 5% from a 322KB average file size and 95% from a 4.1KB average file size from each source (exponentially distributed file size)

- Video Sources per access input link (one of the following):
  - -- 1 FBR Video Source operating at 45 Mbps (A total of 4 sources),
  - -- 12 VBR AR(1) Video Sources operating at 0.50 bits per pixel, 250000 pixels per frame (A total of 48 sources), or
  - -- 29 VBR AR(2) Video Sources operating at 100.71 cells per NTSC frame rate (A total of 116 sources)
- 175 Audio Sources per link operating a FBR of 64k bps (A total of 700 audio sources)

The computer simulation was written in the C++ language and was compiled to run on an IBM compatible personal computer. The simulation was run on various configurations ranging from a 20MHz 386SX based machine to a 25MHz 386DX computer. All computers were populated with a math co-processor. The compiled version of the simulation program was typically less than 600KB and varied depending on the specific network simulation scenario. The compiled version can be run typically on a 286 based machine with 640KB of memory. Typically, the computer simulations were run on machines with at least 2MB of memory to take advantage of the source debugger capabilities included in the compiler software package.

The following is a major list of approximations or adjustments the simulation made in modeling the adaptive DSC algorithm:

- $\text{Queue\_Average\_Size\_threshold}_{Low}$ was set equal to $\text{Queue\_Average\_Size\_threshold}_{High}$
- $T_{Congest}$ = Network's Cell Service Time of 3.68 μsec

REFERENCES

[1] M. Butto, E. Cavallero, and A. Tonietti, "Effectiveness of the Leaky Bucket Policing Mechanism in ATM Networks," IEEE Journal on Selected Areas in Communications, vol. 9, April 1991.

[2] E. Rathgeb, "Modeling and Performance Comparison of Policing Mechanism for ATM Networks," IEEE Journal on Selected Areas in Communications, vol. 9, April 1991.

[3] L. Dittmann, S. Jacobsen, and K. Moth, "Flow Enforcement Algorithms for ATM Networks," IEEE Journal on Selected Areas in Communications, vol. 9, April 1991.

[4] N. Yin and M. Hluchyj, "A Dynamic Rate Control Mechanism for Source Coded Traffic in a Fast Packet Network," IEEE Journal on Selected Areas in Communications, vol. 9, September 1991.

[5] G. Ramamurthy and R. S. Dighe, "Distributed Source Control: A Network Access Control for Integrated Broadband Packet Networks," IEEE Journal on Selected Areas in Communications, vol. 9, September 1991.

[6] B. Maglaris, D. Anastoassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance Models of Statistical Multiplexing in Packet Video Communications," IEEE Transactions on Communications, vol. 36, July 1988.

[7] D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks," IEEE Transactions on Circuits and Systems for Video Technology, vol. 2, March 1992.

[8] P. Sen, B. Maglaris, N. Rikli, and D. Anastassiou, "Models for Packet Switching of Variable-Bit-Rate Video Sources," IEEE Journal on Selected Areas in Communications, vol. 7, June 1989.

[9] H. Heffes and D. M. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," IEEE Journal on Selected Areas in Communications, vol. 4, September 1986.

[10] R. M. Rodriguez-Dagnino, M. Khansari, and A. Leon-Garcia, "Prediction of Bit Rate Sequences of Encoded Video Signals," IEEE Journal on Selected Areas in Communications, vol. 9, April 1991.