Access Policies for Distributed Video-on-Demand Systems

Robert Simon, Padmavathi Mundur and Arun Sood
Department of Computer Science
George Mason University
Fairfax, VA 22030

Abstract

Distributed Video-on-demand (VoD) systems which deliver digitized video directly to users are expected to become one of the most important types of communication technology enabled by the next generation of computer networks, video servers and distributed multimedia systems. This paper discusses some of the challenges posed to computer science in VoD design and implementation. In a VoD system, tight coordination among all subsystems is required for real-time performance. We argue that a central part of any coordination strategy should be end-to-end *threshold-based* admission control policies. End-to-end admission control policies determine if adequate resources are available along the *entire* service path of a new video request so it can be admitted without violating the real-time performance requirements of the request already in service. Threshold-based policies provide a way of assigning higher priority to selected classes of requests based on their relative importance, such as popularity. This is accomplished by imposing a threshold on other classes, thereby restricting access to system resources for those lower priority classes. The paper presents an analytical model to evaluate such threshold-based admission control strategies for distributed VoD systems with multiple classes of video programs. We prove the existence of computationally efficient ways of determining blocking probabilities for newly arriving requests. Through numerical analysis we show that threshold-based policies offer better performance for selected high-priority video classes.

1 Introduction

The rapid evolution of technology and communication protocols for high speed networks coupled with advances in computer, storage, and display hardware and software has paved the way for new types of distributed multimedia applications. These applications allow users great flexibility in deciding *what kind* of information to obtain or exchange, and *when* to obtain or exchange it. This freedom represents a break from traditional types of mass communication using broadcast medium, such as radio or TV, where the providers and content distributors decide what to send and when to send it. The user-centric aspect of distributed multimedia applications represents an entirely new mode of mass communication.

One of the most important types of emerging multimedia applications are distributed Video-on-Demand systems (VoD). A distributed VoD system delivers high quality digitized video directly to viewers. A fully realized VoD system is extremely powerful and flexible because it permits users to decide when, where and what to watch. By untethering viewers from content distributors, VoD offers a prime illustration of some of the creative possibilities offered by distributed multimedia.

Designing and implementing a distributed VoD system poses significant challenges for computer scientists and engineers. This is because, unlike most existing distributed computer communication systems, delivering digitized video to users requires tight coordination of all subsystem components, from files servers and I/O systems, to computer networks, to end-user presentation devices. Such coordination is needed because of the stringent real-time performance and synchronization requirements of video applications and the enormous amount of data present in video objects. A central part of this coordination strategy is the development of an appropriate set of admission control policies. There are two levels of VoD system admission control. The first at the subsystem level, including admission control policies for a video file server. This type of policy may ensure that a request for playing a new video will only be granted if bounds on retrieval latency can be guaranteed without violating the performance requirements of other videos already in service. The second level of admission control policy is end-to-end. End-to-end admission control strategies determine if adequate resources are available throughout the *entire* path of system resources required by the new request so the request can be admitted without violating the continuous playback requirements of the requests already in service.

This paper discusses the design and analysis of a distributed VoD system in the context of developing end-to-end admission control policies. The successful deployment of a widely available VoD system is to a large extent contingent upon the successful development of end-to-end admission control policies and methods. Our focus is on the development an analytical framework to evaluate various priority or *threshold* based admission control. Using this framework we identify appropriate performance measures, such as system throughput under a variety of demand levels, for evaluating the effectiveness of different policies. The results of our analysis provide the necessary techniques with which to perform capacity planning, match user needs to equipment investment, and guide policy evaluation strategies. We also show the existence of efficient quantitative techniques to compute these performance measures. These performance measures are necessary in order to effectively deploy all large scale distributed systems[Menasce et al., 1994].

To make this work concrete, we use the design and implementation of a VoD system on the scale of a hypothetical county-wide school system. While the entertainment industry represents a huge market for future VoD services, we believe that VoD systems can have a critical impact on the quality of services provided by many types of organizations, such as educational institutions, business marketing and training, and internal corporate communications. VoD deployed at the level of a school system can be used to distribute lectures, provide teacher training material and show movies or shorter video clips of educational or cultural value. A major attraction of such a system is its flexibility. Teachers and students select when, where and what kind of videos to watch. Videos can be viewed by larger audiences in a classroom or lecture room setting via TV sets or videos projectors, or individually via desktop and laptop computers. Using the school system as an example, we present numerical analysis which shows the effect of threshold-based admission control on system throughput. Our analysis and numerical results clearly show the need for well designed admission control polices for effectively deploying VoD systems.

The paper proceeds as follows. Section 2 explains how VoD systems could be used by a typical school system. We also describe current and emerging technologies which enable VoD, including issues with respect to video server design and networks, along with compression technology are discussed. Section 3 develops an analytical model for our school VoD system. We prove that efficient algorithms exist for calculating these performance measures. In Section 4, we present numerical analysis to show that threshold-based policies result in lower blocking rates for selected classes of requests, as compared to a naive admission control policy. Section 5 offers some observations and a conclusion.

2 Distributed Video-on-Demand Systems

This section describes system architectures, enabling technologies, and issues in designing admission control policies for distributed video-on-demand systems. These topics are illustrated in the context of a hypothetical VoD system designed to serve the needs of a county wide public school system. We call a system of this type School Video-on-Demand (S-VoD).

2.1 System Architecture

Figure 1 shows the overall S-VoD system architecture. The central repository, or Archival Server (AS), provides high capacity long term video storage. The server is connected via network to school system sites. This network is used to download videos directly to a high speed video server within each site. The videos are downloaded to a particular site upon a user request from that site.

Figure 2 shows the S-VoD distribution system inside a typical school system site. Videos are stored locally on a smaller but faster server, called a Video File Server (VFS). VFSs have network connections to the central AS. A VFS is a high speed file server capable of real-time delivery of multiple streams of digital video. The VFS transmits to video display stations via set-top boxes and PCs, over a Local Area Network. VFSs have lower storage capacities than an AS.

A user requests a particular video via system software. Since PCs and set-top boxes do not have sufficient capacity to store entire video clips, the video is transmitted directly from the VFS into the display device via the school sites LAN. If the requested video is not present on the VFS it needs to be obtained from the AS. An overview of S-VoD operations follows:

- Video clips are stored in compressed form on VFSs and/or at the archive server.

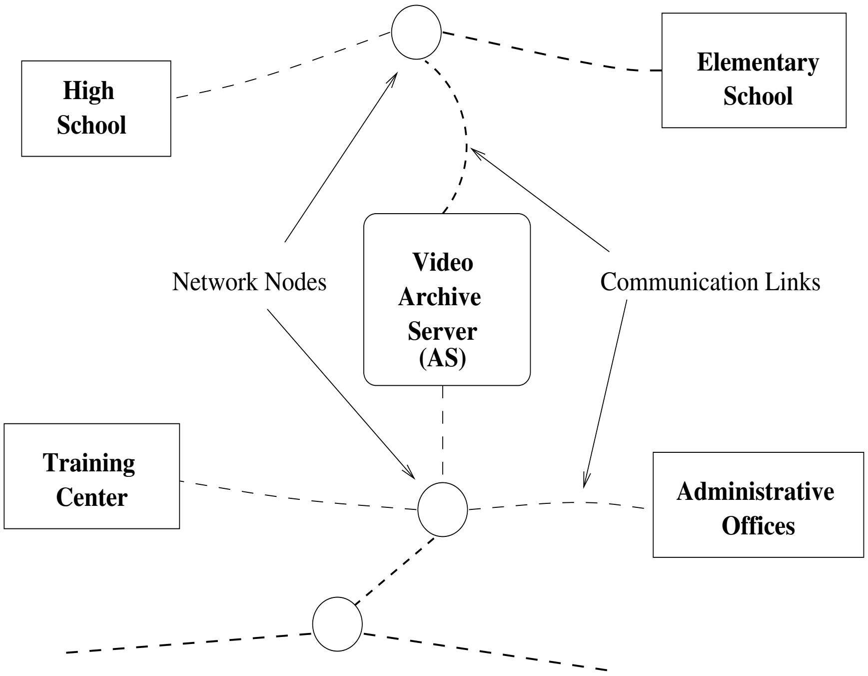

Figure 1: The architecture for a hypothetical Video-on-Demand system for a county wide school district. The system stores videos in a central repository called an Archival Server (AS). Videos are delivered to different locations via a computer communication network.

- An admission control test is performed by the system for each new viewing request. The outcome is either to accept the request and retrieve and display the video, or to reject the request.
- If the video is present on the archive server and not on the local VFS, the video is downloaded to the VFS.
- Once the video data is on the VFS it is continuously retrieved from the VFS and transmitted over the network to the set-top boxes or appropriately-equipped PCs.
- The received video data is decompressed at the set-top boxes and displayed at the receiving sites.

In the next section, we discuss the technologies that enable our S-VoD system such as compression, storage and retrieval at the servers, transmission over networks, and decompression at the set-top boxes.

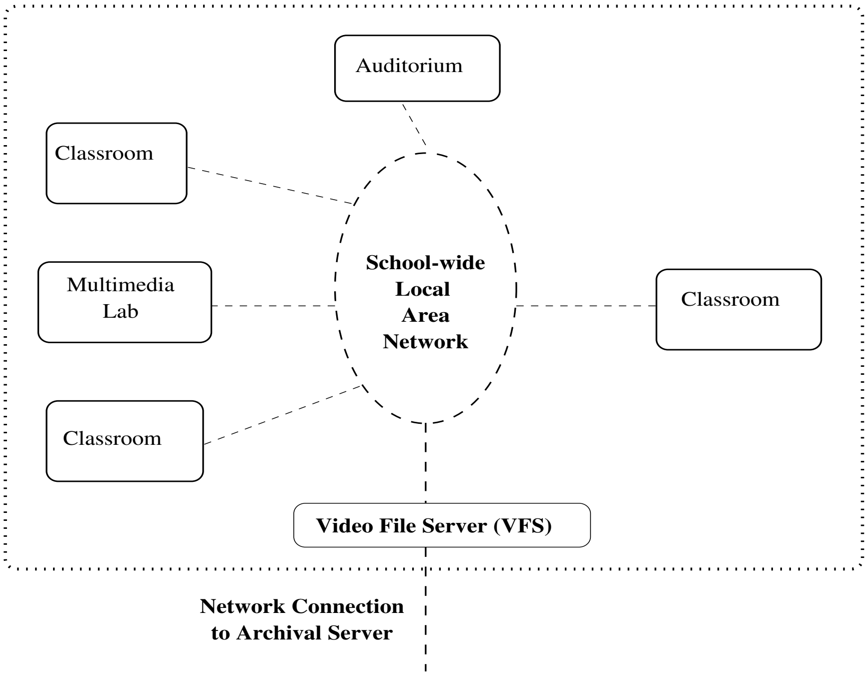

Figure 2: Individual school system sites store local videos for playback on a Video File Server (VFS).

2.2 Enabling Technologies

2.2.1 Compression

Video data is typically stored and transmitted in compressed form. A compression method for color video reduces the amount of space required to digitally represent chrominance and luminance values. This space reduction is usually obtained by several processing steps. Typically, a video frame is divided into square blocks of pixels. This block then undergoes a frequency domain transform step, a quantization step, and a compression step. Further compression is achieved by removing redundant information between frames, e.g., if the background in a scene does not change, there is no need to store the background for each frame. The primary advantage of compression is therefore space savings.

The amount of space savings resulting from compression depends upon the particular technique employed, and can vary from 2 or 3 to 1 to over 100 to 1. Popular compression standards include H.261, for video coding over ISDN lines, different Motion Picture Expert Group (MPEG) standards, such as MPEG-1, MPEG-2 and MPEG-4, and H.263, for very low rate video encoding. Compression standards also vary dramatically in terms of output quality. For instance, the normal resolution of an MPEG-1 frame is 320 by 240 pixels, while for MPEG-2 the resolution is 720 by 480 [Rao and Hwang, 1996]. For our S-VoD system, the compression technique determines the required level of system storage and system throughput. MPEG-1 requires 1.5 Mbps, while data rates for MPEG-2 vary anywhere from 3 Mbps to 100 Mbps.

2.2.2 Video File Servers

Continuous media, such as audio and video, has spatial, temporal and performance constraints not present in textual data retrieval and transfer. Video file servers must offer high storage capacity, low latency retrieval and a high data transfer rate. There are several ways to achieve a cost effective solution to these design problems, including architectural solutions such as faster and larger disks and hierarchical configurations of secondary and tertiary devices that distribute the load within and across different levels [Brubeck and Rowe, 1996].

Server design research has focused on finding efficient storage techniques that minimize the overhead related to the retrieval of multimedia data and maximize the number of concurrent requests. Since disk transfer rates are significantly higher than a single stream real-time data rate, it is possible to serve multiple requests simultaneously. This requires disk head scheduling policies which guarantee continuous and synchronized data retrieval. In particular, it is essential that the disk scheduling policy for data retrieval does not starve, even on temporary basis, one video stream in favor of another. A wide range of disk scheduling algorithms are used to achieve this purpose, including Earliest Deadline First, Scan Earliest Deadline First, Grouped Sweeping Scheme (GSS) and Round-robin. These algorithms typically require an admission control test before the video stream goes into service. Once the request is accepted these scheduling algorithms ensure real-time, continuous and starvation-free retrieval [Gemmell et al., 1995] [Yu et al., 1992].

The effect of requiring admission control for video server access is to place a limit on the number of simultaneous requests that can be handled at the VFS. Different types of admission control tests are possible. For instance, deterministic admission control strategies are based on worst case assumptions regarding service times, while statistical admission control strategies typically use a stochastic characterization for service load and performance reliability [Chang and Zakhor, 1994] [Vin et al., 1994b] [Vin et al., 1994a].

In addition to disk scheduling methods, data placement and retrieval techniques are important, since a multimedia data stream is stored as blocks on a disk. Because of presentation continuity requirements the placement of video data blocks on a disk is critical. This problem is exacerbated by the presence of multiple video streams. Techniques used to solve the block placement problem include *random, contiguous*, and *constrained* block placement[Rangan and Vin, 1993] [Vin and Rangan, 1993]. Use of these policies involves different tradeoffs and considerations. For instance, random block allocation is common in storage server architectures, but may not guarantee support for continuous retrieval, because average or worst-case access and latency times of successive block retrieval may be too large. This forces the use of complex buffer management methods in order to nullify the variation in retrieval time. Contiguous block allocation guarantees continuous retrieval, but results in disk fragmentation and wasted space. Constrained block allocation can be used to satisfy the real-time playback requirements of multiple video streams through block placement and interleaving strategies. These policies also result in simpler buffer management procedures. The drawback is that constrained block allocation policies require detailed knowledge of characteristics of video data for a particular clip or movie.

Hierarchical storage consisting of secondary and tertiary storage devices are essential for meeting large storage requirements of a VoD system. Secondary storage devices at the VFS consists of regular hard disks, RAID devices. While it is expensive to maintain all video programs on secondary storage devices, tertiary storage devices available to each archive server such as magnetic tapes and optical disks provide a cost effective way of storing terabytes of data for a VoD system. These devices are not suitable for directly servicing concurrent user requests because they typically have low data transfer rates.

2.2.3 Networks

The network in a VoD system needs to simultaneously server multiple users. The networked transfer of data between archive servers and the video file servers requires the availability of networks with transmission rates on the order of gigabits per second. These rates are obtainable by high-speed, *Integrated Services Networks* (ISNs) such as Asynchronous Transfer Mode (ATM) or the next generation of the Internet. ISNs provide reliable and real-time data transfer by reserving and allocating resources for connections on an end-to-end basis. Connection establishment protocols include the Resource Reservation Protocol (RSVP) in the Internet, and Q.2931 for ATM [Zhang et al., 1993].

VFSs are connected to end-user display units (described below) by a Local Area Network or other type of distribution service. Examples of these distribution networks include various kinds of Digital Subscriber Line technology using twisted pair wiring for digital transmission, and computer communication networks such as 100 Mbps Ethernet and ATM for a LAN environment.

2.2.4 End-user Display Units

The end-user display unit consists of a high resolution workstation or a high-definition television, an interactive control device, and a network interface. The interactive control device could be a set-top box (STB) providing decoding functionality at the user's premises. The video servers are connected to the STBs over a network. They may also allow some amount of VCR-like interactive control, such as fast forward and rewind. A typical architecture of an STB consists of a *processing subsystem* with an operating system to manage the STB, a *video subsystem* to decompress MPEG-2 type video streams, a *graphics subsystem* for presentation purposes, and an *audio subsystem* to decode audio to synchronize with the video [Furht et al., 1995].

2.3 Threshold-based End-to-End Admission Control

Implementing VoD systems requires the introduction of end-to-end admission control policies. End-to-end admission control strategies determine if adequate resources are available along the path of the new request so it can be admitted without violating the continuous playback requirements of the requests already in service. The path includes both video file servers and communication networks. Threshold-based admission control policies provide a way of introducing priorities among request classes and distinguish between their relative importance. An attractive approach for maintaining certain performance and throughput levels for more popular classes is enforcing a threshold on classes that are not as popular. The threshold relates to the maximum number of unpopular movies which the system will allow to be supported at one time.

2.4 Video Classification

An S-VoD system will only be used if it can meet viewers demands in a flexible manner; be reasonably economical; does not require extensive maintenance; and can accommodate a wide variety of usage patterns. Achieving these goals requires an accurate viewer model. Our admission control analysis models user habits by classifying types of demands for particular videos. One type of classification is *popularity*. In general, VoD systems must support requests for popular videos, i.e., videos repeatedly requested. True VoD systems also must allow access to videos which are relatively unpopular, and are only occasionally requested. Another classification axis is when the video is available for playback. For instance, some videos may be produced and consumed at the same time, similar to broadcast TV. Without loss-of-generality we identify 3 classes of videos which the S-VOD system must support – *archived*, *first-time* and *ad-hoc*. These three user classes enable system builders to design the system to meet anticipated user demand.

Archived videos are videos produced and stored well in advance of their viewing, are popular, and therefore are used in a semi-predictable way. An example is a 45 minute video clip demonstrating orbital mechanics which the school system may have purchased. A high school physics teacher may decide to show this clip at different times during the semester.

First-time videos are videos stored immediately before their usage. Similar to archive videos, first-time videos are popular and are requested in a semi-predictable way. An example of a first-time video may arise during a space shuttle mission. NASA may provide daily status briefings for these missions. Each briefing may last anywhere from 5 to 20 minutes, and can be recorded and distributed digitally to interested school systems via satellite, cable TV or computer network. Individual classes could playback the videos that day during homeroom periods.

Ad-hoc videos are videos whose usage is relatively spontaneous and whose average usage patterns cannot reasonably be predicted in advance. They may or may not be stored (or produced) well in advance of their usage. An example of this kind of clip is a high school English teacher who wants to show an experimental production of a Shakespearian play as an ad-hoc response to a particularly interesting classroom discussion about alternate staging.

3 S-VoD Analytical Model

3.1 Threshold-based Admission Control

An admission control procedure for all arriving requests is run at the VFS. The procedure determines if capacities are available on the VFS and the network to transfer the data to the user display devices. If the program needs to be downloaded from the archive server, the availability of network capacity for that transfer is checked. In addition to the capacity constraints, the VFS assesses further class-based constraints imposed by the admission control policy.

An admission control *policy* determines whether an arriving request for a certain class of videos should be admitted, given the current state of the S-VoD system. Threshold-based policies provide the means for increasing the number of videos from one class the system can support. For example, in our situation we may need to place access restrictions on ad-hoc requests (class 3), so we can support more of the semi-predictable requests (class 1 and class 2). A threshold *class-3* policy limits the class 3 requests up to a specified threshold, and always accepting class 1 and class 2 requests if there is available capacity.

Without loss of generality, suppose there are three classes of requests and the vector (n_1, n_2, n_3) describes the state of the S-VoD system in terms of the number of class 1 requests, n_1, class 2 requests, n_2, and class 3 requests, n_3. The state of the S-VoD system is describable as a state space constrained by capacity. A capacity constraint is defined by $\sum_{k=1}^{3} n_k \leq C$ where C is the maximum number of simultaneous requests that a VFS can handle. The state space can be further constrained by other restrictions, such as when the admission control policy restricts access to one the other classes by enforcing a threshold on one of the classes. The state space for a threshold class-3 policy is illustrated in Figure 3. The state space is bounded by the three positive co-ordinates and the triangular plane representing the capacity limitation. The threshold plane on the class 3 dimension truncates the state space. All the admissible states are within this bounded space. The state space, Ω, is defined by

$$\Omega \equiv \{(n_1, n_2, n_3) : 0 \leq n_1 \leq C \text{ and } 0 \leq n_2 \leq C \text{ and } 0 \leq n_3 \leq l_3 \text{ and } \textstyle\sum_{k=1}^{3} n_k \leq C\}$$

Given the state space of the system based on the admission control policy, we can identify those states where an arrival is blocked either because of the capacity constraint or the threshold. We compute blocking probability using this information.

3.2 Three Class Server Model

Each VFS at a school site has access to a centrally located pool of archive servers. The video programs will be served only through the VFS. The programs from the archive servers, therefore need to be first downloaded to the VFS. The VFS handles a maximum of C simultaneous requests, primarily determined by the I/O bandwidth of the server. The VFS acts as C virtual servers, each of which is serving a request. We assume that the requests for viewing new videos arrive according to a Poisson distribution and the service

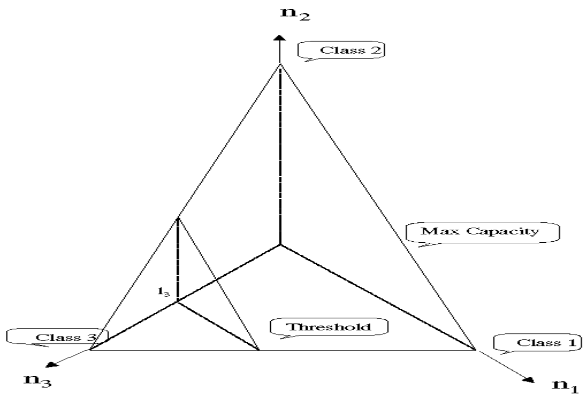

Figure 3: S-VoD State Space

times according to a general distribution. The Poisson assumption for arrival rates is the standard assumption used in analysis of VoD queueing systems [Li et al., 1996].

The state of the S-VoD system is describable by a Markov chain with a birth-death process and a state space Ω. By using the Markov chain model, the threshold type admission control policy falls into a class of policies known as *coordinate convex* [Aein, 1978]. Arrivals or departures are never blocked, except for boundary states, where any further arrivals are blocked. The advantage of coordinate convex policies is that they give rise to product form solutions to the equilibrium state probabilities, from which performance measures such as blocking probabilities can be determined.

There are three classes of requests: class 1 refers to archival programs, class 2 to first-time programs and class 3 to ad-hoc programs. Based on this model class 1 and class 2 requests are more "important" then class 3, and more of those requests should be admitted into the system. We impose a threshold on class 3 requests because we would like class 1 and class 2 requests to have access to a larger share of the server resources.

We assume adequate network resources between the VFS and the user display devices, as well as to the archive servers. This means that there is no blocking at either the network or the archive server level. The VFS is modeled as a single node, multi-server, 3 class queueing model. The following analytical model evaluates admission control policies based on different threshold values for class 3 at the video server. Blocking probability for each class under a certain class 3 threshold value is computed using the approach below. We first present some model parameters and definitions:

Model Parameters:

λ_1	average arrival rate of request for archival programs (class 1)
λ_2	average arrival rate of request for first-time programs (class 2)
λ_3	average arrival rate of request for ad-hoc programs (class 3)
$1/\mu_1$	average service time for class 1
$1/\mu_2$	average service time for class 2
$1/\mu_3$	averate service time for class 3
C	maximum number of requests that the VFS can handle simultaneously
l_k	threshold for class k, $k = 1, 2, 3$

Definitions:

- P_{b_1} - blocking probability for class 1 requests
- P_{b_2} - blocking probability for class 2 requests
- P_{b_3} - blocking probability for class 3 requests
- $P(n_1, n_2, n_3)$ - probability of n_1 class-1 requests and n_2 class-2 requests and n_3 class-3 requests at the VFS, $\sum_{k=1}^{3} n_k \leq C$
- Ω - set of admissible states under a given policy

  for a system with threshold on all classes,

  $\Omega \equiv \{(n_1, n_2, n_3) : 0 \leq n_1 \leq l_1$ and $0 \leq n_2 \leq l_2$ and $0 \leq n_3 \leq l_3$ and $\sum_{k=1}^{3} n_k \leq C\}$

  for a threshold class-3 system,

  $\Omega \equiv \{(n_1, n_2, n_3) : 0 \leq n_1 \leq C$ and $0 \leq n_2 \leq C$ and $0 \leq n_3 \leq l_3$ and $\sum_{k=1}^{3} n_k \leq C\}$
- Ω_1 - set of blocking states for class 1 requests
- Ω_2 - set of blocking states for class 2 requests
- Ω_3 - set of blocking states for class 3 requests

As shown in [Aein, 1978] and [Kaufman, 1981], the equilibrium probability, $P(n_1, n_2, n_3)$, has the following product form:

$$P(n_1, n_2, n_3) = \frac{1}{G} \prod_{k=1}^{3} \rho_k^{n_k}/n_k!, \quad \text{all } (n_1, n_2, n_3) \in \Omega \tag{1}$$

where

$$G = \sum_{(n_1,n_2,n_3) \in \Omega} \prod_{k=1}^{3} \rho_k^{n_k}/n_k!$$

and

$$\rho_k = \lambda_k / \mu_k$$

The blocking probability, P_{b_k}, is given by

$$P_{b_k} = \sum_{(n_1,n_2,n_3) \in \Omega_k} P(n_1, n_2, n_3) = G(\Omega_k)/G \tag{2}$$

where $\Omega_k = \{(n_1, n_2, n_3) | (n_1, n_2, n_3) \in \Omega, \ (n_1, n_2, n_3) + \overline{e_k} \notin \Omega\}$ and $\overline{e_k}$ is a vector set to $(1, 0, 0)$, $(0, 1, 0)$ or $(0, 0, 1)$, where the 1 is in the k^{th} position. Ω_1, Ω_2, and Ω_3 are the set of blocking states for class 1, class 2, and class 3 requests respectively.

The cardinality of Ω grows on the order of C^k, making exhaustive enumeration techniques computationally too expensive. Hardware limitations rule out computing G for even moderate sized problems. We therefore must develop alternate, computationally efficient methods for computing G. Some previous studies in finding computationally tractable solutions for computing G were done in the context of multi-rate circuit switched networks. For instance, Kaufman [Kaufman, 1981] and Roberts[Roberts, 1981] independently developed recursive relations to compute the G factor for complete sharing policies. Threshold type policies introduce additional blocking states that should be included in the computation of blocking probabilities. Also, the state space Ω is now truncated according to the threshold values. Kaufman/Roberts recursion was generalized to compute blocking probability for multi-rate tree networks in [Tsang and Ross, 1990]. Here we develop a recursion for a single node, multi-server, 3 class threshold type system that is similar to the recursion presented in [Tsang and Ross, 1990] [Mundur et al., 1998]. The recursion is applicable to one or more classes having a threshold restriction.

The basic idea behind the recursion is summing $P(n_1, n_2, n_3)$ over Ω recursively along disjoint parallel planes on the diagonal, as shown in Figure 4 for a two-class situation. Since each request requires one server at the VFS, the number of requests present in the system also indicates the number of server resources being used. From the summing technique, we get a distribution in terms of the total number of servers being used. Denote the total number of servers used as $I = \sum_{k=1}^{3} n_k$, and define a state space,

$$\Theta(i) = \{(n_1, n_2, n_3) \in \Omega : \sum_{k=1}^{3} n_k = i\}$$

The distribution q_i, indicates the number of servers being used.

$$q_i = P(I = i) = \sum_{(n_1,n_2,n_3) \in \Theta(i)} P(n_1, n_2, n_3), \quad i = 0, 1, \ldots, C \tag{3}$$

Define,

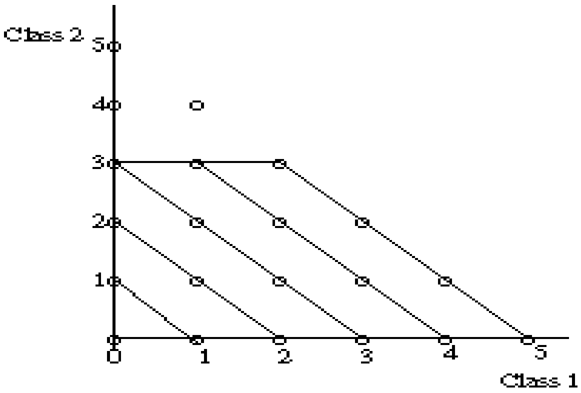

Figure 4: Planar States for Distribution q_i

$$G(i) = \begin{cases} \sum_{(n_1,n_2,n_3)\in\Theta(i)} \prod_{k=1}^{3} \rho_k^{n_k}/n_k! & i = 0, 1, \ldots, C \\ 0 & \text{otherwise} \end{cases}$$

Multiplying (3) with G and substituting for $P(n_1, n_2, n_3)$ from (1) we get the expression above for $G(i)$,

$$G(i) = q_i G \tag{4}$$

To handle threshold type policies, we define a threshold value which is less than C to limit the number of requests for those classes. Define $l_k < C$ as the threshold for class k requests. Define for $k = 1, 2, 3$,

$$B_k(i) = \begin{cases} P(I = i, n_k = l_k)G & \text{if } l_k \leq i \leq C - 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$B_k(i)$ is the unnormalized probability of being in one of the additional blocking states due to the threshold. If no threshold is defined for a particular class, $B_k(i)$ for that class is 0. The blocking probability P_{b_k} is given by:

$$P_{b_k} = P(I = C) + P(I \leq C - 1, n_k = l_k)$$

$$P_{b_k} = \frac{G(C) + \sum_{i=l_k}^{C-1} B_k(i)}{\sum_{i=0}^{C} G(i)} \tag{6}$$

To compute blocking probability, we need only calculate the G factor and $B_k(i)$, $i = l_k, \ldots, C - 1$, for $k = 1, 2, 3$. We use the following recursion to accomplish that.

Theorem 1

$$G(i) = \frac{1}{i}\sum_{k=1}^{3} \rho_k [G(i-1) - B_k(i-1)], \quad i = 1, \dots, C \tag{7}$$

The proof of Theorem 1 is contained in the Appendix. The result is the computationally efficient form we have been seeking.

Based on our classification of viewer needs, it is reasonable to limit class 3 ad-hoc requests and allow more of predictable class 1 and class 2 requests to be accepted into the system. We therefore must compute $B_k(i)$ for a threshold class-3 system. A threshold class-3 system allows class 3 requests up to a threshold and always accepts class 1 and class 2 requests if capacity is available. By definition $B_1(i) = 0$ and $B_2(i) = 0$. A new function $r(i)$ is used to compute $B_3(i)$ and to determine the resource distribution of class 1 and class 2 requests. This is defined as

$$r(i) = \frac{1}{i}[\rho_1 r(i-1) + \rho_2 r(i-1)], \quad i = 0, \dots, C - 1 - l_3 \text{ and } r(0) = 1$$

$r(i)$ is similar to $G(i)$ and is computed using the same recursion as in Theorem 1, but only on class 1 and class 2 state space. Notice that by definition, $B_3(i)$ indicates a resource usage of l_3 requests for class 3 .

$$B_3(i) = (\rho_3^{l_3}/l_3!)\, r(i - l_3), \quad i = l_3, \dots, C - 1 \tag{8}$$

Stripping off the recursion in (8), we have

$$B_3(i) = (\rho_3^{l_3}/l_3!)[\rho_1^{(i-l_3)}/(i-l_3)! + \rho_2^{(i-l_3)}/(i-l_3)!]$$

which can be proved from equilibrium conditions, the product form expression in (1), and the definition of $B_3(i)$.

4 Numerical Results

This section demonstrates the effect of various threshold-based admission control policies on blocking for each of the classes. We show that using carefully chosen threshold levels for threshold-based admission control policies results in higher acceptance rates for selected classes. Specifically, the acceptance rates are higher for selected classes for threshold policies than complete sharing policies, where all classes of requests are admitted into the system as long as capacity is available. An admission control policy defines a threshold for class 3 and admits class 3 requests only up to that threshold. Different threshold values result in different levels of blocking for each class. The acceptable blocking level for each class is decided based on the performance goals of the VoD system. The numerical results test the advantage of having flexible class 3 thresholds and the ability to set thresholds at various levels to achieve the desired performance for all classes.

Our experimental setup studies the effects of changing arrival rates and threshold values. The average service times are set to one. The capacity of our hypothetical school S-VoD system is assumed to be $C = 500$, allowing up to 500 simultaneous requests. This could roughly match the number of network connections a county-wide school system might have. Further, this capacity should be large enough to accommodate requests from various classes and laboratories in the school for video on demand. We assume a dedicated VFS for each school. Within the school environment, we assume that there is enough network bandwidth to all the display stations. We assume no blocking due to the network or the archive server wherever archival transfers are considered.

For our S-VoD system, we assume three classes of requests, class 1 (archival), class 2 (first-time), and class 3 (ad-hoc) to the VoD system. We assume that class 1 and class 2 requests are more important than class 3 requests and should receive higher priority and a larger share of the server resource. For that reason, we impose a threshold restriction on class 3, restricting the sharing of server resources. One result of this policy is that even if there is server capacity available in the S-VoD system, a class 3 request is blocked by the threshold restriction. Class 1 and class 2 requests are accepted as long as there is capacity available. Since both class 1 and class 2 are limited by the capacity, blocking probability will be the same for both.

We use the results of the previous section to compute blocking probability for each class. The steps involved are:

- The recursion $r(i) = \frac{1}{i}[\rho_1 r(i-1) + \rho_2 r(i-1)]$ is used to first compute all $r(i)$s.
- Equation (8) is used to compute $B_3(i)$ next.
- Once all $B_3(i)$ are determined, Theorem 1 is used to compute $G(i)$.
- Blocking probability for all three classes are calculated as follows:

$$P_{b_1} = \frac{G(C)}{\sum_{i=0}^{C} G(i)} \tag{9}$$

$$P_{b_2} = \frac{G(C)}{\sum_{i=0}^{C} G(i)} \tag{10}$$

$$P_{b_3} = \frac{G(C) + \sum_{i=l_3}^{C-1} B_3(i)}{\sum_{i=0}^{C} G(i)} \tag{11}$$

- The throughput is computed using $\lambda_k(1 - P_{b_k})$.

A computer program was written to compute blocking probabilities using equations (9), (10) and (11). The program used standard 15 digit double precision using GNU C

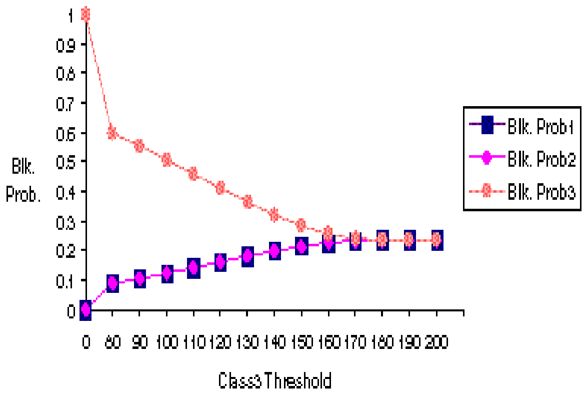

Figure 5: Blocking Probabilities on the y-axis for varying Class 3 Threshold on the x-axis,ρ_1 = 300, ρ_2 = 150, ρ_3=200

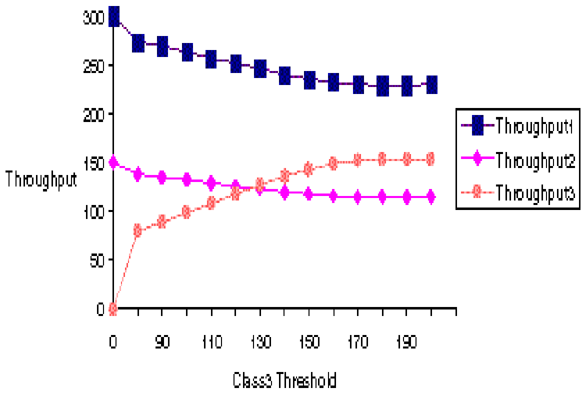

Figure 6: Throughput on the y-axis for varying Class 3 Threshold on the x-axis, ρ_1 = 300, ρ_2 = 150, ρ_3=200

compiler on a Sparc workstation to compute our results. All numerical experiments assume a maximum capacity C of 500 simultaneous requests. The main focus of threshold-based analysis and these experiments is to set a threshold that results in a tolerable level of blocking for class 3 with a desirable level of blocking for class 1 and 2.

The results of the first numerical analysis are shown in Figure 5, demonstrating the effect of varying the admission threshold for class 3 while ρ_1, ρ_2, and ρ_3 are held constant. The threshold for class 3 ranges from 80 to 200, ρ_1 is set to 300 and ρ_2 to 150 and ρ_3 to 200. As the threshold for class 2 approaches the value of ρ_3 the blocking probability for class 1 increases as a result of increased sharing of resources. The blocking probability for class 3 decreases because of increased threshold. Eventually the situation becomes similar to complete sharing policy, where each class has the same blocking probability. Notice that the blocking probability for both class 1 and class 2 are the same because they

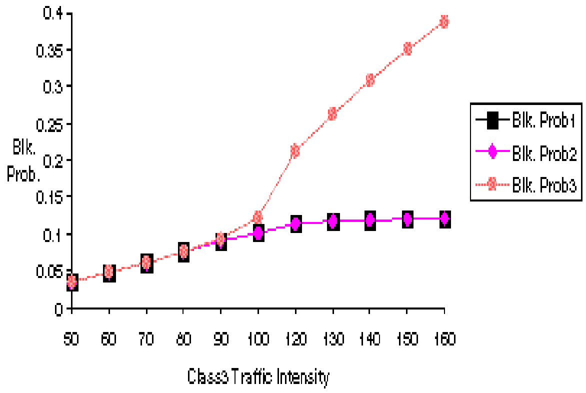

Figure 7: Blocking Probabilities on the y-axis for varying Class 3 Traffic Intensity on the x-axis,ρ_1 = 300, ρ_2 = 150

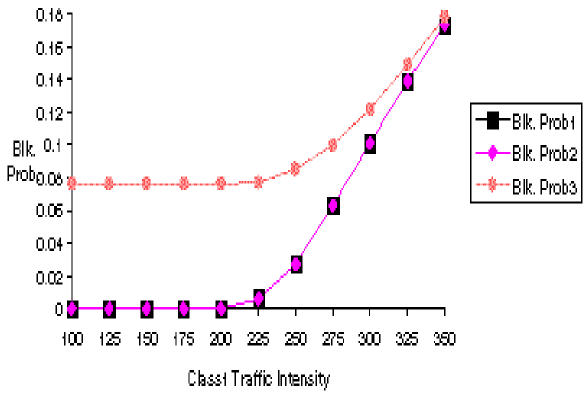

Figure 8: Blocking Probabilities on the y-axis for varying Class 1 Traffic Intensity on the x-axis, ρ_2 = 150, ρ_3=100

are both limited by the capacity of the VFS. However, the respective throughput for class 1 and class 2 will depend on the arrival rate for each class. The throughput using the same set of experimental parameters is shown in Figure 6.

Figure 7 shows the effect of increasing the arrival rate of class 3 while maintaining a fixed threshold at 100. The blocking probability sharply increases for class 3 as the arrival rate goes beyond the threshold, while the blocking probability for class 1 and class 2 barely varies.

Figure 8 shows the effect of increasing the arrival rate for class 1, while maintaining a fixed arrival rate for class 2 and 3 and a fixed threshold for class 3 at 100. Blocking for both classes remains constant until the value of ρ_1 approaches capacity, at which point, blocking for both classes increases rapidly, as expected. At higher values of class 1 traffic intensity, all three classes have the same blocking as capacity becomes the overriding limiting factor for all three classes and threshold effects become negligible.

The results shown in Figures 5 – 8 demonstrate the basic relationship between the arrival rates of classes, the threshold of class 3, and the resulting blocking probability. The advantage of being able to change thresholds on class 3 to achieve the desired blocking on class 1 and 2 is illustrated in Figure 9. Lower thresholds on class 3 result in lower blocking probability for class 1 generally and in particular, keep blocking to a low level in the face of increasing class 1 arrival rate.

Figure 10 shows that low thresholds again result in stable blocking for class 1 and class 2 as class 3 arrival rates are increased. There is more variability in class 1 and 2 blocking with respect to increasing class 3 arrival rate at higher thresholds. This result is expected because at higher thresholds, the situation degrades to one of complete sharing.

The effect of varying class 1 and class 3 arrival rate in lock step on class 3 blocking is shown in Figure 11. Low thresholds result in higher blocking for class 3, more so at higher traffic intensities and as mentioned before, results in lower blocking and stable performance for class 1 and class 2.

All of these results demonstrate the advantage of using threshold based end-to-end admission control policies in order to manage system throughput for popular videos.

5 Conclusion

In this paper, we discussed issues related to the deployment of a distributed VoD system. We argued that video-on-demand systems represent a new type of user-driven communication technology. We presented a hypothetical video-on-demand system for a county-wide school system, called S-VoD. We described enabling technologies required for deploying a S-VoD system. We also provided an analytical model to evaluate the performance of such VoD systems under various admission control policies.

By running a series of numerical experiments we showed that threshold-based analysis results in better performance and lower blocking for class 1 and class 2 requests than if all three classes had the same access to the server resources at the VFS. Our numerical results show that threshold-based admission control polices are an appropriate and effective tool for our S-VoD system with multiple classes of requests, each with differing

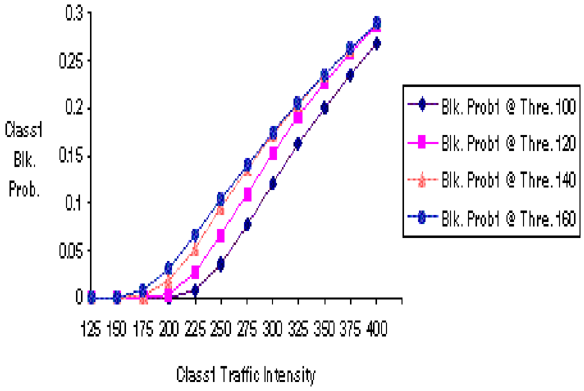

Figure 9: Class 1 Blocking Probability on the y-axis for varying Class 1 Traffic Intensity at different Thresholds on the x-axis, ρ_2 = 150, ρ_3= 150

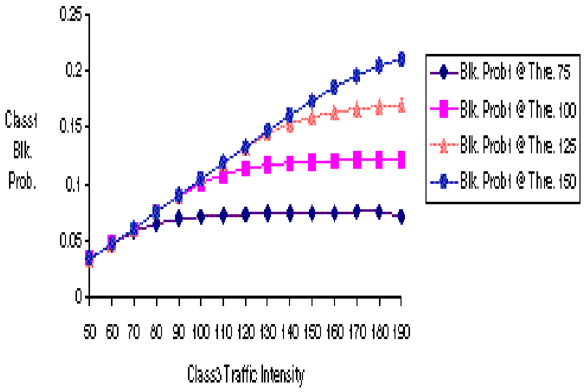

Figure 10: Class 1 Blocking Probability on the y-axis for varying Class 3 Traffic Intensity at different Thresholds on the x-axis, ρ_1 = 300, ρ_2 = 150

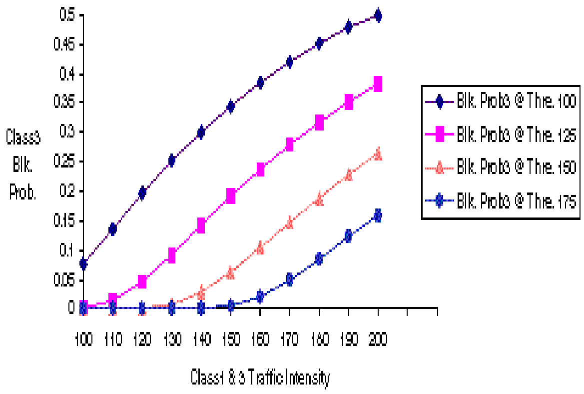

Figure 11: Blocking Probabilities on the y-axis for varying Class 1 and 3 Traffic Intensity at various Thresholds on the x-axis, ρ_2 = 150

importance. The results of this analysis can be used to guide the deployment of a real Video-on-Demand system on the scale we have described.

References

[Aein, 1978] Aein, J. M. (1978). A multi-user-class, blocked-calls-cleared demand access mode. *IEEE Transactions on Communications*, COM-26(3):378–385.

[Brubeck and Rowe, 1996] Brubeck, D. W. and Rowe, L. A. (1996). Hierarchical storage management in a distributed vod system. *IEEE Multimedia*, pages 37–47.

[Chang and Zakhor, 1994] Chang, E. and Zakhor, A. (1994). Admission control and data placement for vbr video servers. In *Proceedings of the 1st IEEE International Conference on Image Processing*, pages 278–282. IEEE Press.

[Furht et al., 1995] Furht, B., Kalra, D., Kitson, F. L., Rodriguez, A. A., and Wall, W. E. (1995). Design issues for interactive television systems. *IEEE Computer*, 28(5):25–39.

[Gemmell et al., 1995] Gemmell, D. J., Vin, H. M., Kandlur, D. D., Rangan, P. V., and Rowe, L. A. (1995). Multimedia storage servers: A tutorial. *IEEE Computer*, 28(5):40–49.

[Kaufman, 1981] Kaufman, J. S. (1981). Blocking in a shared resource environment. *IEEE Transactions on Communications*, COM-29(10):1474–1481.

[Li et al., 1996] Li, V. O. K., Liao, W., Qiu, X., and Wong, E. W. M. (1996). Performance model of interactive video-on-demand systems. *IEEE Journal on Selected Areas in Communications*, 14(6):1099–1109.

[Menasce et al., 1994] Menasce, D., Almeida, V., and Dowdy, L. W. (1994). *Capacity Planning and Performance Modeling: from mainframes to client-server system*. Prentice Hall.

[Mundur et al., 1998] Mundur, P., Sood, A., and Simon, R. (1998). Threshold-based admission control for multi-class video-on-demand systems. In *Proceedings of the IEEE International Performance, Computing, and Communications Conference*, pages 154–160.

[Rangan and Vin, 1993] Rangan, P. V. and Vin, H. M. (1993). Efficient storage techniques for digital continuous multimedia. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):564–573.

[Rao and Hwang, 1996] Rao, K. and Hwang, J. (1996). *Techniques and standards for image, video, and audio coding*. Prentice Hall.

[Roberts, 1981] Roberts, J. W. (1981). A service system with heterogenous user requirements - application to multi-services telecommunications systems. In *Performance Data Communications Systems Conference Proceedings*, pages 423–431. North-Holland Publishing Co.

[Tsang and Ross, 1990] Tsang, D. H. K. and Ross, K. W. (1990). Algorithms to determine exact blocking probabilities for multirate tree networks. *IEEE Transactions on Communications*, 38(8):1266–1271.

[Vin et al., 1994a] Vin, H. M., Goyal, A., Goyal, A., and Goyal, P. (1994a). An observation-based admission control algorithm for multimedia servers. In *Proceedings of the First International Conference on Multimedia Computing and Systems*, pages 234–243.

[Vin et al., 1994b] Vin, H. M., Goyal, P., Goyal, A., and Goyal, A. (1994b). A statistical admission control algorithm for multimedia servers. In *Proceedings of the ACM International Conference on Multimedia*, pages 33–40.

[Vin and Rangan, 1993] Vin, H. M. and Rangan, P. V. (1993). Designing a multiuser hdtv storage server. *IEEE Journal on Selected Areas in Communications*, 11(1):153–164.

[Yu et al., 1992] Yu, P., Chen, M., and Kandlur, D. D. (1992). Design and analysis of a grouped sweeping scheme for multimedia storage management. In *Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 38–49.

[Zhang et al., 1993] Zhang, L., Deering, S., Estrin, D., Shenker, S., and Zappala, D. (1993). Rsvp: A new resource reservation protocol. *IEEE Network*, 7(5):8–18.

Appendix: Proof of Theorem 1

Let $\overline{e_k}$ be $(0,0,1)$ or $(0,1,0)$ or $(0,0,1)$, where the 1 is in the k^{th} position. Under equilibrium conditions, we have

$$\sum_{(n_1,n_2,n_3)\in\Theta(i)} P((n_1,n_2,n_3)-\overline{e_k}) = P(I=i-1) - P(I=i-1, n_k=l_k) \tag{12}$$

Consider resources used $I=\sum_{k=1}^{3} n_k$ and multiplying (3) with i and substituting for i and rearranging on the RHS, we have

$$iq_i = \sum_{k=1}^{3} \sum_{(n_1,n_2,n_3)\in\Theta(i)} n_k P(n_1,n_2,n_3)$$

Substituting from the balance equation, $n_k P(n_1, n_2, n_3) = \rho_k P((n_1, n_2, n_3) - \overline{e_k})$ we have,

$$iq_i = \sum_{k=1}^{3} \sum_{(n_1,n_2,n_3)\in\Theta(i)} \rho_k P((n_1, n_2, n_3) - \overline{e_k})$$

$$iq_i = \sum_{k=1}^{3} \rho_k \sum_{(n_1,n_2,n_3)\in\Theta(i)} P((n_1, n_2, n_3) - \overline{e_k})$$

Substituting from (12) for $\sum_{(n_1,n_2,n_3)\in\Theta(i)} P((n_1, n_2, n_3) - \overline{e_k})$ we have,

$$iq_i = \sum_{k=1}^{3} \rho_k [P(I = i - 1) - P(I = i - 1, n_k = l_k)]$$

Multiplying with G and using (4) and moving i to RHS we have the recursion in Theorem 1.

$$G(i) = \frac{1}{i} \sum_{k=1}^{3} \rho_k [G(i - 1) - B_k(i - 1)] \quad i = 1, \ldots, C$$

and $G(i) = 1$ for $i = 0$ and 0 for negative i.

QED