Content-based VBR Video Traffic Modeling and its Application to Dynamic Network Resource Allocation

Paul Bocheck [†] *and Shih-Fu Chang*
{bocheck,sfchang}@ctr.columbia.edu

Department of Electrical Engineering and New Media Technology Center
Columbia University, New York, N.Y. 10027-6699, U.S.A.
tel. (212) 939-7155, fax: (212) 316-9068

Abstract

In bandwidth limited networks and network interfaces, dynamic resource allocation can substantially increase the link utilization and also decrease the required network buffering. In general, there are two important tradeoffs effecting the network utilization. First, tradeoff between efficiency of the real-time dynamic resource allocation and the policy controlling the renegotiation frequency. Second, tradeoff between accuracy of stream resource prediction and prediction delay. The latter is the focus of our work. We propose a new *Content-based Video Traffic Model* which uses the visual content as an important indicator of the video stream resource requirements. The model has two components: the visual content is characterized by *Video Structure Model* and the particular compression mechanism is described by *Scene Resource Model*. The *object-based video content classification scheme*, first introduced in our work, maps individual video scenes into their bandwidth resource requirements. We validate our classification scheme for the MPEG-2 stream using a real 0.5 hour VBR MPEG-2 video trace. Also, using a trace-driven network simulator we show main advantages of our new technique: improvement of network utilization. From the results we find that while the performance of off-line content-based dynamic network resource allocation scheme shows marginal improvement compared to existing approaches (e.g., RVBR), its on-line performance shows significant improvements (55% - 70%).

Keywords: VBR video traffic modeling, video content classification, dynamic network resource allocation

1 Introduction

Recent advances in media technology, stimulated by high processing power and large storage have drastically increased demands for heterogeneous, high bandwidth communications. In the near future, networks are expected to carry diverse multimedia traffic such as video, audio and data with a full range of quality of service (QoS). Such networks require efficient resource management which is challenged by tradeoff between the network utilization and QoS guarantee. The higher network utilization can be achieved by network resource sharing and statistical multiplexing. However, to avoid the network congestion, call admission and flow control must be enforced.

[†]corresponding author

Because of constant picture quality, support of VBR video in high speed packet networks is desirable. Unfortunately, due to its complicated traffic characteristics, effective transport of real-time VBR video with guaranteed QoS has been a challenging issue. Recent studies show that, if not handled properly, VBR video traffic may lead to network congestion or a very low network utilization [1].

One promising solution, aimed to overcome these difficulties, is to replace the *static resource allocation* with *dynamic resource allocation*. In the static resource allocation the network resources are allocated only once at the beginning of the session. On the contrary, the dynamic resource allocation supports in-call resource renegotiation. The network resources, based on current need, are dynamically requested while the session is in progress. Compared to the static one, dynamic resource allocation may vastly increase the network utilization. It is most efficient when traffic model is unknown, model parameters are difficult to obtain or static allocation is not efficient.

While the general principle of dynamic resource allocation is straightforward, the major challenge remains in the selection of renegotiation strategies, especially in the case of real-time VBR video. The solution appears to be connected to traffic prediction. Since many published traffic prediction models were based solely on a single prediction indicator (bit-rate), their performance and accuracy was limited [4, 5, 6]. In this paper we propose a new traffic prediction scheme for use in dynamic resource allocation based on the *Content-based Video Traffic Model*. Its design was motivated by correlation between the video structure (content) and its compressed representation(bit-rate). In this paper we extend our previous content-based model to accommodate more general video features [11]. We show that the exploration of these features helps us to predict VBR video stream resources more accurately.

The rest of the paper is organized as follows. In Section 2 we briefly discuss open questions related to the congestion control and transport of VBR video streams over the packet networks. We examine network topology and show under which conditions the dynamic resource allocation may increase the external link and network utilization. In the reminder of the paper we focus on content-based video traffic modeling and its application to the dynamic network resource allocation. In Section 3, we describe a new Content-based Video Traffic Model. We discuss, in general, a temporal structure and spatial features of video documents (movies, real-time broadcasts, etc.) and their relationship to video compression. Based on our observations we identify video content and compression mechanism as two independent components which can be modeled separately. The *Video Structure Model* is able to capture the most important visual properties (content) of video while the video traffic resulting from various compression mechanisms is modeled by *Scene Resource Model*. In the context of dynamic resource allocation we develop a *content-based video segmentation algorithm* and *object-based scene classification scheme* relating the visual properties of the video to the compressed video traffic. We experimentally confirm this connection using the real MPEG-2 stream. In Section 4 we examine the use of our content-based model in real-time dynamic resource allocation. In the last Section 5 we experimentally show its superior performance by comparing the results of trace-driven simulation of content-based model to results obtained by using existing models.

2 Network resource allocation

In this section we outline network management and resource allocation issues directly related to network utilization. In particular we discuss conditions under which dynamic resource allocation, based on the three network layer congestion control model, can substantially increase the network throughput.

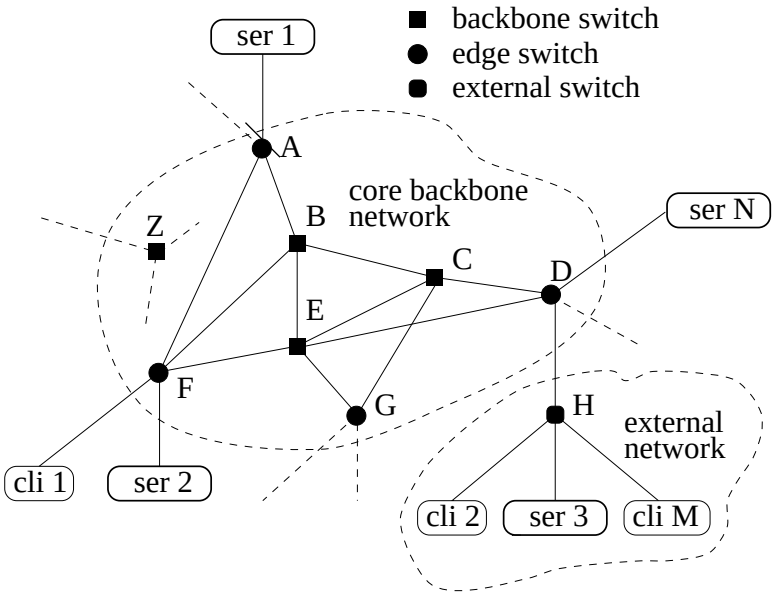

Figure 1: Network topology.

Figure 1 depicts a typical topology of packet network. In this example the network consists of multiple servers $ser1, ..., serN$ and clients $cli1, ..., cliM$ interconnected through nodes $A, B, ..., Z$. We place no restrictions on interconnection nodes, servers and clients; the interconnection node can be ATM switch or high speed packet router, the server could be a digital camera or video server while client could be a video set-up box, workstation or wireless multimedia terminal.

According to traffic pattern, the networks can be logically partitioned into core (backbone) and external networks [2]. The external network is a part of the user networking infrastructure. It has two primary functions: to provide the access to the core network and to serve as a local area network (LAN). The primary function of the backbone network is to carry the aggregated traffic flowing between the external networks. Note that traffic conditions in the external and core networks substantially differ. For example, ATM core network includes very high capacity links between powerful ATM switches which are able to process thousands of virtual circuit connections. On the other hand, the external network switches and network interfaces have typically lower bandwidth capacity and are able to process only limited number of connections. To provide efficient transport of VBR video streams, these distinct traffic conditions must be taken into the consideration. As a result, network management should be able to support variety of resource allocation, congestion control, and call admission strategies.

The transport of video over the packet networks is connected with the following important issues. Because of high burstiness and the multiple-time scale property of VBR traffic, in order to obtain reasonable statistical multiplexing gain, the number of multiplexed streams could be rather large [9]. Unfortunately, this requirement cannot be satisfied in bandwidth limited external networks. Also, because the VBR bursts of high bit rate could occur for a relatively long time, to allow transmission at lower than peak rate, large network buffers associated with large delay would have to be used to assure the QoS. Under these conditions, the traditional two layer (cell and call admission) resource allocation is inefficient (requires near peak rate bandwidth allocation) and leading to very low network utilization. To keep the delay small and still provide required

QoS, controlled access to the network resources on the burst layer is necessary. For networks or network interfaces, where the link capacity is not able to accommodate large number of streams, more flexible multilayer congestion control model for transport of bursty traffic was suggested in [3]. In this model the congestion is controlled at three layers: packet (cell), burst and call. For example, at the call level control, the new call is denied if the excessive burst blocking probability occurs; similarly the burst of the already admitted call is denied if it would cause excessive cell loss at the cell layer. This model is characterized by increased granularity of control leading to higher network utilization. Assumptions of similar traffic conditions makes this model directly applicable to transport of VBR video in the external networks. In the backbone network, it should be sufficient to operate in two layers (namely, cell or packet level and call admission) only. To provide guaranteed QoS, both external and core networks must cooperate in the network resource management and the call admission.

Dynamic resource allocation is based on three layer congestion control model. It is a technique allowing network resources to be allocated on need bases during the lifetime of the connection. Request for the change of resources is generated when source traffic conditions change; for example, at the beginning of excessive cell bursts. As it will be explained later, while request for decrease of resources will always be granted, the request for increase can be denied. Such scheme creates a possibility of statistical multiplexing at the burst layer while providing an important functionality of congestion control: individual stream protection. Increase of statistical multiplexing is an important factor directly related to the increase of network utilization.

The efficiency of dynamic resource allocation depends on strategy of determining the renegotiation intervals and selection of the appropriate *traffic descriptor*. The selection strategy is discussed in more detail in the Section 3. The renegotiation intervals are found by *video segmentation* algorithms, dividing video frame sequence into variable size sections. While there exists optimal offline segmentation algorithms, currently proposed renegotiation strategies for real-time VBR video are inadequate. One possible way to improve the efficiency is to be able to predict the video traffic more precisely. Since many published traffic prediction models are based solely on a single prediction indicator (bit-rate) without looking into the content structure of the video stream, their performance and accuracy is limited [4, 5, 6].

We propose our *Content-based Video Traffic Model* to improve the prediction of the resources required by real-time video resulting in efficient dynamic resource allocation. As we will show later, content-based approach takes into account the real process of VBR stream generation, providing the basis for more accurate traffic prediction and efficient video segmentation ultimately resulting in an increase of network utilization. Also, the content-based video segmentation operates on a desirable time scale of several seconds (corresponding to the scene length scale), which may avoid the bottleneck caused by the excessive request messages and computation demands for dynamic resource allocation.

3 Content-based Video Traffic Model

Figure 2 depicts a 1000-frame-long segment of VBR MPEG-2 encoded video stream. The full trace, used in our trace-driven simulations, is 54000 frames long ($\approx$ 0.5 hour) and was created using Columbia University's MPEG-2 software encoder from the movie *Forrest Gump*. The trace illustrates the central idea of content based traffic modeling: the correlation between visual content and the corresponding bit rate. To better visualize the trend of I, P, and B frames we use frame envelope curves: frame envelope connects the same frame types (see legend). The vertical dotted lines in Figure 2 mark changes in visual content of the video (such as appearance of new scene,

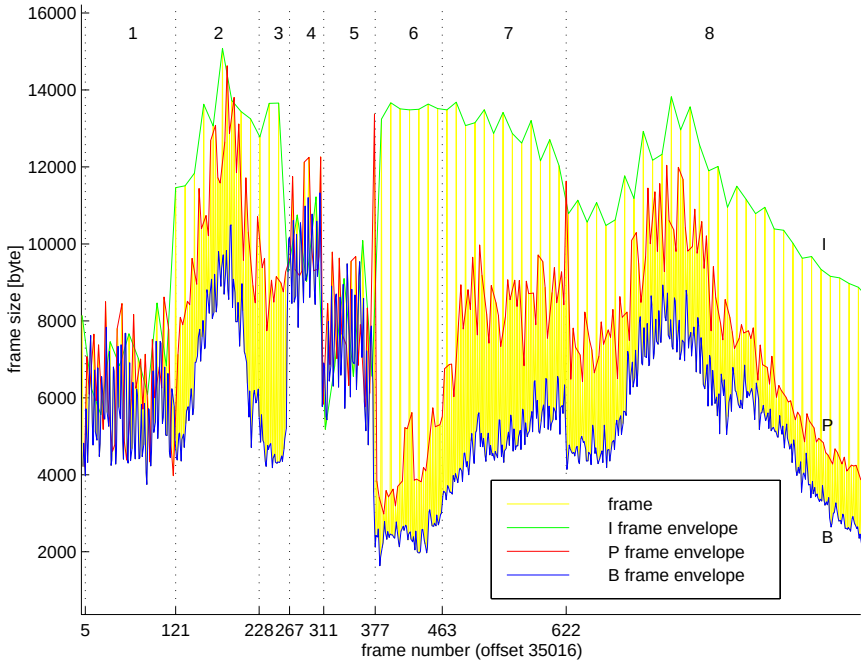

Figure 2: MPEG-2 VBR trace, movie Forrest Gump.

beginning/end of camera motion, sudden object movement, etc.). Note, that these changes coincide with visually observed discontinuities in the trace.

By visual inspection, we can identify MPEG-2 VBR video stream as a discrete periodic stochastic process with bursty and non-stationary behaviors. This video trace can be better understood once we take into consideration its visual content. Using a simple scene classification, we can show that different scene types (classes) have, in general, distinct traffic pattern. For example, segment 1 is a scene with smooth background and high speed camera panning in horizontal direction. The same high speed horizontal camera panning appears at the scenes corresponding to the segments 4 and 5, although both these scenes are not as smooth as in the segment 1 (e.g. medium smoothness or complexity). Segment 2 corresponds to the cluttered (high complexity) scene with medium speed camera panning in the vertical direction. Segments 3 and 6 are both static scenes (e.g. without a camera motion) with cluttered background; segment 3 corresponds to the scene with a single large (secondary) object moving with the medium speed while scene in segment 6 has no moving objects. Both segments 7 and 8 are scenes with cluttered background; additionally, the segment 7 corresponds to the scene with medium speed camera zooming while segment 8 corresponds to the scene with high speed camera zooming.

Using the previous example, we can infer, without loss of generality, that the stream resource requirements are influenced by the complete video production process (video content creation) and also the video compression. This observation leads us to a key concept of content-based video traffic modeling: the dependence of video compression rate on two independent components: *visual content* and *compression mechanism*. We call this decomposition the *separation principle* [11]. For example, different video encoders might produce traffic streams with fundamentally different characteristics while still carrying the same visual content. The separation principle is schematically shown in Figure 3. It depicts the Content-based Video Traffic Model partitioned into two independent parts: (1) *Video Structure Model* and (2) *Scene Resource Model*.

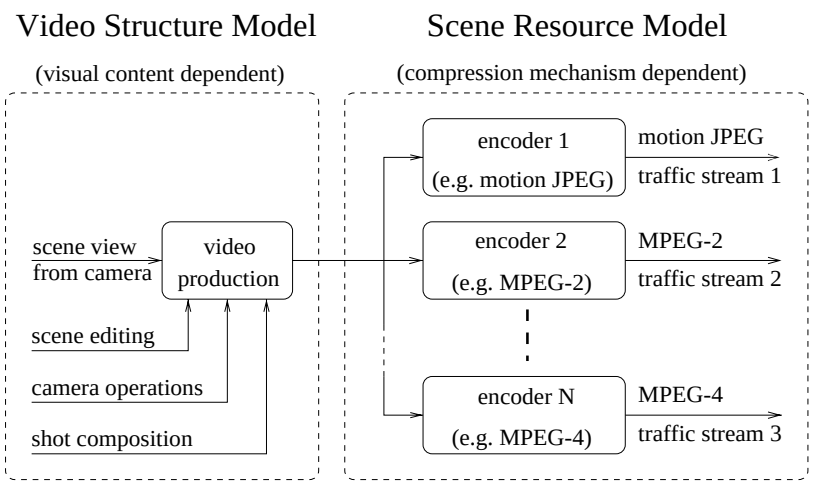

Figure 3: Separation principle in the Content-based Video Traffic Model.

The Video Structure Model corresponds to the video production process including the shot composition, scene editing, the control of the video camera, etc. It characterizes video at the content level: first by segmentation of the video into the scenes, followed by segmentation of scenes into video objects. We will show later how video scenes and objects can be classified by various visual features: for example by their size, spatial complexity and motion.

The Scene Resource Model is sensitive to compression mechanisms. Its function is to map the scene visual content into the traffic descriptors. It has been shown that traffic characteristics corresponding to individual scenes do not possess extreme and complicated behaviors (e.g. non-stationarity) and may be modeled by stationary Markov or AR(n) models [14]. In other words, each scene, can be expressed by relatively simple traffic descriptor. These results are not surprising when we take into consideration the general property of the scene: scene is typically perceived as a time interval or collection of frames in which visual content does not substantially change. As we will show later, the use of relatively simple traffic descriptor greatly simplifies structure of the Scene Resource Model. Both the Video Structure Model and the Scene Resource Model are discussed in more detail in Section 3.1 and Section 3.2 respectively.

The separation principle allows us to supply another dimension into the video traffic modeling. Our goal, in the rest of the paper, is to show, that the scene resource requirements can be inferred directly from the scene content itself.

3.1 Video Structure Model

A Video Structure Model has two main components. The first one deals with *video segmentation*: decomposition of the video (such as movies, news, live broadcast programs, etc.) into the scenes. The second one deals with *scene segmentation*: decomposition of each individual scene into the set of video objects and their characterization.

A video $V = \{s_i\}_{i=1}^{m}$ can be perceived as a collection of m *scenes* s_i. In general, the scenes depict some real world scenery or actions. With the availability of advanced image processing and editing technology, video scenes can also be synthesized, enhanced, digitally manipulated, or contain various special visual effects. A scene can be defined in many ways. Typically, it is defined as a video segment between two distinct camera shots. Our definition is more general: a scene

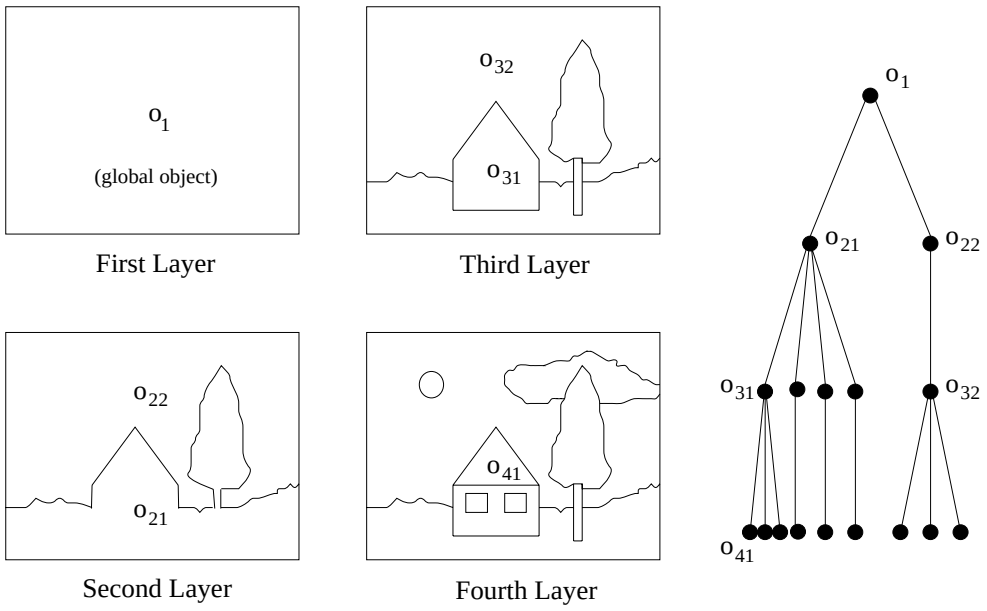

Figure 4: Hierarchical object-based scene segmentation scheme.

is a video segment during which the visual content does not substantially change. In that sense, the single camera shot can be segmented into several scenes, depending on its content (i.e. camera movement, object movement, etc.). For example, at the beginning of the shot, camera may be static (first scene), then new object may appear on the scene (second scene) and finally the camera may follow the moving object (scene three). The algorithm for partitioning a video stream into segments of different content is called *content-based video segmentation* and will be discussed in more detail in Section 3.4.

The *video object* description is based on general and very flexible *hierarchical object-based scene segmentation scheme* [15]. We adopted and enhanced this technique to be suitable for description and classification of video scenes and video objects in the context of video traffic modeling. Depending on the required segmentation accuracy (layer of decomposition), video scene can be segmented into several smaller video objects. Example of the hierarchical scene segmentation is depicted on Figure 4. At the first layer, the whole image is considered as a single global object. With increasing resolution, more objects are identified. For example, at the second layer, there are two objects. First object, o_{21} is associated with the foreground (house, tree, etc.), second object o_{22} is associated with background (sky). Similar hierarchical decomposition applies at higher levels. The important feature of the hierarchical object-based scene segmentation scheme is its compatibility with the video object decomposition in the proposed MPEG-4 coding system.

3.1.1 Scene content description

Generally, the scene content can be characterized at two ways: the first one identifies the scene type (global features) while the second one describes the objects, appearing on the scene [11]. The scene type is described in terms of *scene type descriptor (STD)* and video objects are described in terms of *video object descriptors (VOD)*. In other words, the scene s_i, consisting of several video

objects $\{o_1, o_2 \cdots\}$ can be described as follows:

$$s_i \longrightarrow \{STD(s_i), VOD(o_1), VOD(o_2), \cdots\} \tag{1}$$

where $STD(s_i)$ is the scene type descriptor of the scene s_i and $VOD(o_k)$ is the video object descriptor of object o_k.

The scene type descriptor characterizes scene according to the global features affecting the entire scene and all video objects in the scene. For example, a scene can be described according to the camera movement (*i.e. camera static, panning, translation, and rotation*) or internal camera operations (i.e. *zooming*). The number of different scene types is not fixed and generally depends on intended applications. In the case of MPEG-2 video traffic modeling, discussed in more detail in Section 3.3.2, we found it is sufficient to differentiate scenes into three types of camera operations: (1) camera static, (2) camera panning and (3) camera zoomming. In that case we define scene type descriptor as follows:

$$STD(s_i) \stackrel{\triangle}{=} SceneType(s_i) \qquad SceneType(s_i) \in \{static, panning, zooming\} \tag{2}$$

The video object descriptor characterizes individual video objects in the scene. According to hierarchical object-based scene segmentation scheme, described previously (refer to Figure 4), a scene $s_i = \{o_j \mid j = 1, 2, ...N\}$ is composed of single object (global) or several video objects o_j (primary, secondary, etc.). We define the video object o_j as a spatially segmented image region having consistent features such as color, texture and/or motion. There are many techniques available for segmenting scenes to separate video objects [17, 18]. These segmentation techniques usually are based on combination of visual features such as motion, edge and color. Although the segmented regions may not correspond to real objects perfectly, they have been applied successfully to video coding and indexing [12, 15, 19]. Similarly, we argue that automatic scene segmentation algorithms are satisfactory for our work of traffic modeling, in which accurate mapping of segmented objects to real physical objects is not necessary.

Each video object o_j, as it appears on the scene, has its specific features. For example, objects can be of different size, shape, complexity (cluttered vs. smooth texture) and can move at various speeds and directions. Note, that while some of these features have a direct influence on object bit allocation, others do not (e.g. motion direction, etc.). In general, only selected video object characteristics, strongly related to the video object resource capacity (bit allocation) are considered and included in the video object descriptor $VOD(o_j)$. In other words, the use of the specific video object characteristics depends on the compression technique. We found, in our MPEG-2 experiments, three video object features appear to be more important in terms of their influence on the resulting video stream resource requirements. Therefore, we included them in the video object descriptor. They are: (1) *object size* $S(o_j)$, (2) *object spatial complexity* $C(o_j)$, and (3) *relative object speed* $M(o_j)$:

$$VOD(o_j) \stackrel{\triangle}{=} \{C(o_j), M(o_j, S(o_j))\} \tag{3}$$

3.1.2 Estimation of video object features

In the estimation of video object features included in the video object descriptor we should take into consideration the following observations. First, since many currently in use and future* compression schemes are block based, it would be to the advantage to use the same black-based structure in the

*MPEG-4 uses the block based structure only partially.

video object feature estimation. Second, possibility of feature estimation in the compressed domain carries out a considerable performance advantage in terms of processing time and delay because images do not need to be converted back to the spatial (original) domain.

The spatial complexity $C(o_j)$ is defined using the concept of the rate-distortion function. Rate-distortion functions represent the relationship between the rate allocation and the resulting distortion. Higher image complexity is generally associated with higher rate-distortion values. Under some approximations we may, in case of hierarchical scene decomposition, assume an independence of the video object features. In that case, the rate-distortion function of the composite scene can be derived directly from rate-distortion function of individual video objects.

In reality, to create independent frequency components, most of the encoders use the block-based transformations which are then quantized and entropy coded. For example, blocks of size 8 pixels x 8 pixels are used for discrete cosine transformation (DCT) in JPEG and MPEG-1/2/4 encoders. With some approximation errors, we could view each frequency component as a discrete i.i.d. process X with zero mean and variance σ^2. Under the criterion of small to medium distortion (compared to standard deviation), the entropy of an independent coefficient can be approximated by the following formula [16]:

$$B(\Delta) = \frac{1}{2}\log_2(12\epsilon^2\frac{\sigma^2}{\Delta^2}) \tag{4}$$

where σ^2 is the variance of the process X, Δ is the quantizer step size, and ϵ is source model dependent constant equal to about 1, 1.2, and 1.4 for Uniform, Laplacian and Gaussian distribution (pdf) of the process X respectively. Then, the average rate-distortion function of the MxM block size would be:

$$\overline{B}(\mathbf{\Delta}) = \frac{1}{M^2}\sum_{i=0}^{M^2-1} B_i(\Delta_i) = \frac{1}{2M^2}\sum_{i=0}^{M^2-1}\log_2(12\epsilon_i^2\frac{\sigma_i^2}{\Delta_i^2}) = \frac{1}{2M^2}\log_2\prod_{i=0}^{M^2-1}(12\epsilon_i^2\frac{\sigma_i^2}{\Delta_i^2}) \tag{5}$$

where $B_i(\Delta_i)$ denotes entropy of coefficient i of the block, Δ_i denotes quantization step size for coefficient i and $\mathbf{\Delta}$ denotes matrix of quantization step sizes for block.

Equation 5 suggests to estimate the complexity in two ways. First, from the variance of the frequency components (i.e. σ^2) or directly from the entropy of each of frequency coefficients. Because our goal is to evaluate the object descriptors in the compressed domain, we define complexity c_k of the block b_k using the latter measure: as the sum of the entropy of each DCT coefficient inside the block. In practice, we estimate the entropy of the DCT coefficient by using the number of bits $\hat{B}_i$ used for that coefficient:

$$c_k \triangleq \sum_{i=0}^{M^2-1} B_i(\Delta_i) \cong \sum_{i=0}^{M^2-1}\hat{B}_i \tag{6}$$

Then, the spatial complexity of the object is defined as an average complexity of all blocks belonging to the same object:

$$C(o_j) \triangleq \frac{1}{\sum_{k=0}^{N-1} 1_{\{b_k\in o_j\}}}\sum_{k=0}^{N-1} 1_{\{b_k\in o_j\}}c_k \tag{7}$$

where N is a total number of blocks in the picture.

Object speed $M(o_j)$ can be evaluated in two ways. First, in the original spatial domain as a relative speed of the object. Second, in the compressed domain as an average motion vector computed over all blocks belonging to the same object. Since the latter can be conveniently

extracted from the video stream when encoders support the block-based motion compensation, we define object motion as follows:

$$M(o_j) \stackrel{\triangle}{=} \frac{1}{\sum_{k=0}^{N-1} 1_{\{b_k \in o_j\}}} \sum_{k=0}^{N-1} 1_{\{b_k \in o_j\}} |\vec{m}_k| \tag{8}$$

where $\vec{m}_k$ is a motion vector of the block b_k.

The object size $S(o_j)$ is defined as a number of transformed DCT blocks belonging to the same object.

$$S(o_j) \stackrel{\triangle}{=} \frac{1}{N} \sum_{k=0}^{N-1} 1_{\{b_k \in o_j\}} \tag{9}$$

Note that although we used some parameters specific to the MPEG compression standard to define video object features included in the video object descriptor, these definitions can be easily replaced by other features computable in the original pixel domain. The use of MPEG-specific parameters allow us to extract these features directly from input compressed video streams without converting videos back to the uncompressed pixel domain. Due to the simplicity we have chosen deliberately, video content classification can be done in real time for live video as well.

3.2 Scene Resource Model

Despite a very large number of VBR video traffic models proposed over the last couple years, no model was flexible enough to be effectively used in different applications in which various types of real video traffic need to be modeled [8]. It appears that the selection of the appropriate traffic model, the structure and the complexity is influenced by the intended application. Several models directly applicable to dynamic resource allocation have been recently proposed. Besides the peak rate and effective bandwidth allocation models [5], the resource bounding models were also suggested for the dynamic resource allocation [6]. In the following we focus on the *deterministic bounding interval dependent model* (D-BIND) for scene resource requirements.

Rather than modeling the traffic stream at the single fixed time interval (e.g. frame) basis, the D-BIND resource bounding model predicts stream's bandwidth over several different time scales. There are many advantages in using this model in dynamic resource allocation. First, it greatly simplifies the model construction process, especially when traces such as those in the MPEG-2 video are considered. Second, we found in the context of video content modeling the D-BIND model is a very good compromise between model complexity and its ability to effectively describe the content dependent VBR video characteristics. Third, the source bounding model has the ability to characterize the source burstiness over different time scales. Fourth, the deterministic D-BIND traffic model demonstrated that peak rate allocation is not necessary in order to provide the deterministic QoS guarantee for the VBR traffic [6]. Consequently, the D-BIND model achieves higher network utilization. Finally, traffic conformity to the deterministic-bound source descriptors can be easily policed using a number of leaky bucket regulators. These were the main reasons for selection of the D-BIND resource bounding model as a traffic descriptor of our video content-based model in this paper. However, our proposal using the separation principle in the content-based model and its application to resource renegotiation can be applied in general cases independent of specific traffic model used.

Following is a brief description of the D-BIND model [10]. Denote $A[\tau, \tau + t]$ a cumulative arrivals of the source s during the interval $[\tau, \tau + t]$. Define the empirical envelope $B^*(t)$:

$$B^*(t) \stackrel{\triangle}{=} \sup_{\tau > 0} A[\tau, \tau + t] \qquad \forall t > 0 \tag{10}$$

The empirical envelope $B^*(t)$ represents the tightest time-invariant bound on the source arrivals for every interval $[\tau, \tau + t]$ of length t. Define a family of traffic constrained functions $\mathcal{B}$:

$$\mathcal{B} \stackrel{\triangle}{=} \{B(t) \mid B^*(t) \leq B(t)\} \qquad \forall t > 0 \tag{11}$$

We say that the source s is deterministically bounded by traffic constrained function $B(t)$ when $B(t) \in \mathcal{B}$. In other words, for source arrivals during the time interval of length t holds:

$$A[\tau, \tau + t] \leq B^*(t) \leq B(t) \qquad \forall t, \tau > 0 \tag{12}$$

The D-BIND model refers to a parameterized continuous traffic constrained function $B_{W_T}(t) \in \mathcal{B}$ defined on a set of P time intervals $T = \{t_k\}_{k=1}^{P}$ in terms of pairs $W_T = \{(q_k, t_k) \mid k = 1, 2, ..., P\}$:

$$B_{W_T}(t) \stackrel{\triangle}{=} q_k + \frac{q_k - q_{k-1}}{t_k - t_{k-1}}(t - t_k) \qquad t_{k-1} \leq t \leq t_k \tag{13}$$

with assumption of $q_0 = 0$ and $t_0 = 0$. In other words, the set of points W_T defines $B_{W_T}(t)$ as a continuous piece-wise linear function bounding the empirical envelope $B^*(t)$ from above. We refer to the traffic constrained function $B_{W_T}(t)$ as a D-BIND traffic constrained function.

There could be different ways to construct $B_{W_T}(t)$, but the following procedure can be used to construct $B^*_{W_T}(t)$, representing a tight bound on $B^*(t)$. The procedure computes, for a given empirical envelope $B^*(t)$ and time intervals $T = \{t_k\}_{k=1}^{P}$, the values of q_k such that $W_T = \{(q_k, t_k) \mid k = 1, 2, ..., P\}$ defines the D-BIND traffic constrained function $B^*_{W_T}(t)$. The algorithm is as follows:

1. Input: $B^*(t)$ and $T = \{t_k\}_{k=1}^{P}$
2. Initialize starting point $Y_0 = \{q_0 = 0, t_0 = 0\}$
3. For $k = 1$ to P
4. Find $Y_k = \{q_k, t_k\}$ corresponding to minimum q_k such that line $\overline{Y_{k-1}, Y_k}$ is never below $B^*(t)$ on time interval (t_{k-1}, t_k).
5. end
6. Output: $W_T = \{Y_k \mid k = 1, 2, ..., P\}$

Figure 5 depicts schematically cumulative arrival function $A(0, t)$, empirical envelope $B^*(t)$ and two D-BIND traffic constrained functions $B_{W_T}(t)$ and $B^*_{W_T}(t)$. The $B^*_{W_T}(t)$ represents optimal (tight) D-BIND traffic constrained function. The $B_{W_T}(t)$ is another possible D-BIND traffic constrained function, which is not tight.

It is sometimes more convenient to express D-BIND traffic constrained function as a rate-interval pairs $R_T = \{(r_k, t_k) \mid k = 1, 2, ..., P\}$ such that $r_k = q_k/t_k$ denotes bounding rate over the interval of length t_k. In the rest of the paper we refer to this specification as at D-BIND traffic resource descriptor.

The D-BIND traffic resource descriptor (R_T)determines the bounds on the stream traffic resource requirements used in dynamic resource allocation. In off-line content-based video segmentation, the exact D-BIND traffic resource descriptor (bandwidth requirements) for all scenes can be readily obtained directly from the video stream. More complicated is obtaining D-BIND traffic resource requirements for the live video in real time, discussed in more detail in Section 4.

3.3 Object-based scene classification scheme

As we have already mentioned, the distinguished feature of the content-based resource allocation is effective prediction of the video stream bandwidth requirement utilizing the visual content of the

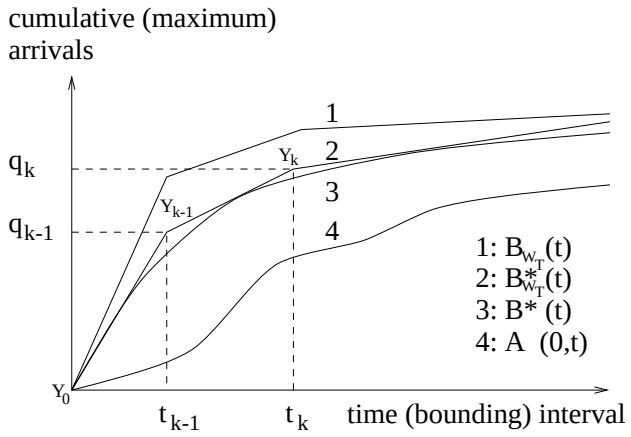

Figure 5: $A(0,t)$, $B^*(t)$ and $B_{W_T}(t)$ traffic functions.

video. The idea of exploitation the video content in traffic modeling is relatively new, however, it is a natural extension of the separation principle, described at the beginning of Section 3.1 (Figure 3). It is based on the analysis of video content structure and its relation with video compression. The content-based approach allows more accurate resource prediction while keeping the model complexity at the manageable level.

There are several key observations, directly leading to the use of content in the traffic modeling. During the encoding, the spatial-temporal visual features of the video are mapped into the compressed bit stream. As we will show later, the video scene content can be approximately characterized with just a few key visual features which are directly related to the stream resource requirements. This relationship is very important and it is a key point on which the content-based video scene classification is based. However, the exact form of such relationship appears to be very complicated to be expressed in the closed form. Consequently, we propose the object-based scene classification scheme.

The scene classification scheme is designed to simplify the content-based resource mapping. It can be expressed in the following way. Denote scene s_i content class as C_i and traffic resource requirements as R_i. Then content-based resource mapping $\mathcal{M}$ can be expressed as follows:

$$\mathcal{M} : C_i \rightarrow R_i \tag{14}$$

In other words, the traffic resource requirements of the video stream are determined directly from the scene class.

The scene classification scheme is scalable in three dimensions. First, it uses the multi-layer object-based scene segmentation hierarchy. At each segmentation layer, more precise spatial object segmentation is performed. With the new advanced image segmentation algorithms, it is possible to recognize and characterize video objects at certain levels in real time[†] [16, 17, 18]. Second, various set of key visual features (e.g. camera operations; object complexity, motion, size, etc.) are used as key components in the scene classification scheme. With the assumption of the video object independence, the object-resource relationship can be easily extended from frame-based features to object-based features. Finally, to optimize the efficiency of the classification, each key feature is quantized independently from each other.

[†]With the accuracy adequate for traffic modeling since the perfect correspondence with real-world objects is still difficult.

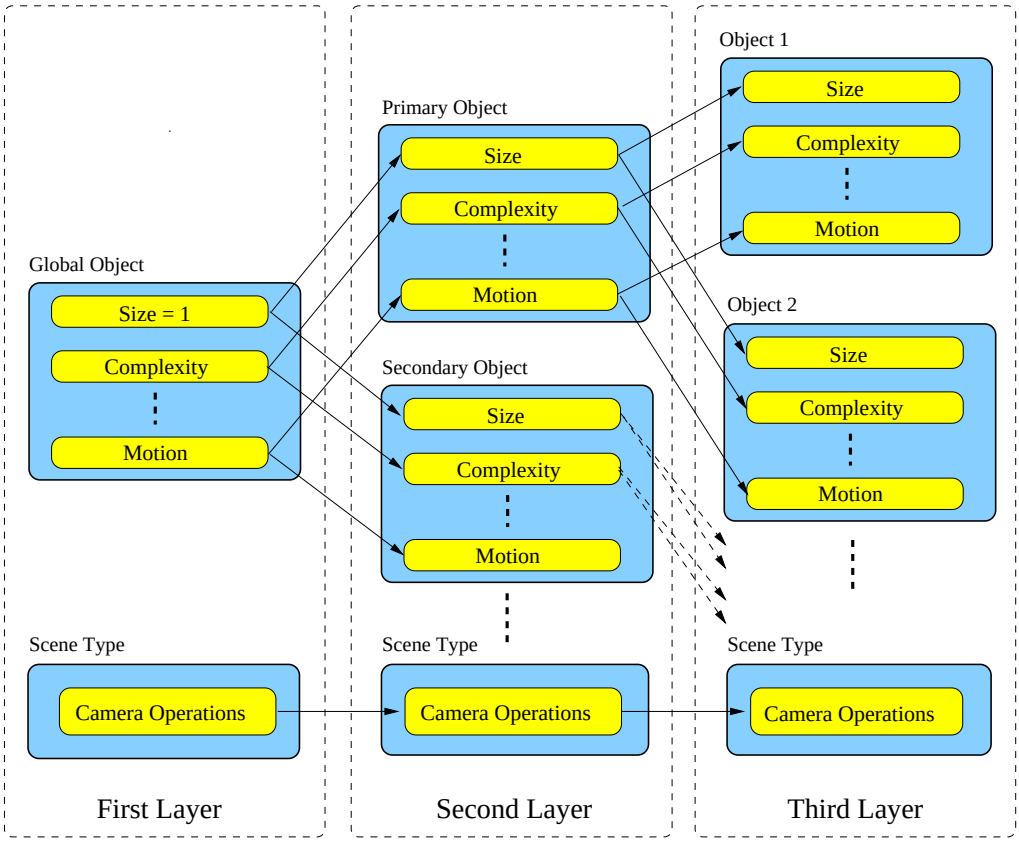

Figure 6: Object-based scene classification scheme.

Based on our previous discussion, we propose the following *object-based scene classification scheme*, depicted in Figure 6. It is directly based on the Video Structure Model, described in Section 3.1, and is structured according to hierarchical scene content segmentation layers (as seen in Figure 4). At each layer, the set of video objects, together with their visual features and quantization levels, defines the scene classification models. An example of such family of models, used for MPEG-2 video classification using two video objects, is depicted on Figure 7.

3.3.1 Evaluating the classification scheme

To verify our theory of close relationship between video scene content and traffic resource requirement, we need an effective measure of the consistency of traffic characteristics among video scenes classified into the same class. Ideally, an accurate mapping process should map video scenes of the same content type to "similar" traffic descriptors. Therefore, the "distance" measure among traffic descriptors of video scenes in the same class should be small.

Since we are interested in the classification scheme for the dynamic network resource allocation, we have chosen the traffic descriptor as a base on which the accuracy of the classification will be measured. For this purpose we use the D-BIND traffic resource descriptor, representing the bounds on the scene traffic in terms of P interval-bandwidth pairs $R_T = \{(r_k, t_k) \mid k = 1, 2, ..., P\}$. This descriptor can also be rewritten as a vector $\vec{X} = \{x_1, x_2, \cdots, x_P\}$, where x_k represents the bandwidth r_k corresponding to the interval t_k. In general, the vector $\vec{X}$ can also define a point X in the P-dimensional space $\mathcal{S} \equiv \mathcal{S}^P$. We use the distance between the points in $\mathcal{S}$ in determining

the degree of consistency of resource descriptors. The square distance between two points X and Y in the space $\mathcal{S}$ can be written as:

$$\Delta^2(X,Y) = (y_1 - x_1)^2 + (y_2 - x_2)^2 + \cdots + (y_P - x_P)^2 \tag{15}$$

Further assume, that each point X, representing the scene, can be classified according to various features. We denote $\mathcal{P} = \bigcup_{n=1}^{N} p_n$ as a set of N partitions, each representing a specific scene class. Note that partitions are mutually exclusive and holds $p_i \cap p_j = \phi$ for $i \neq j$. In other words, each point $X \in \mathcal{S}$ is classified and assigned into a single partition p.

We define *within class distance* $\mathcal{D}_{in}(p)$ and *between class distance* $\mathcal{D}_{out}(p)$ in the following way:

$$\mathcal{D}_{in}(p) = \frac{1}{2} \sum_{X_i, X_j \in p} \Delta^2(X_i, X_j) \qquad \mathcal{D}_{out}(p) = \sum_{X_i \in p, X_j \notin p} \Delta^2(X_i, X_j) \tag{16}$$

Using Equation 16 we define two measures on the partition set $\mathcal{P}$, namely the *degree of grouping* $G(\mathcal{P})$ and *degree of separation* $S(\mathcal{P})$:

$$G\{\mathcal{P}\} \triangleq \frac{1}{\mathcal{D}_{in}(\mathcal{P})} \sum_{p_k \in \mathcal{P}} \mathcal{D}_{in}(p_k) \qquad S\{\mathcal{P}\} \triangleq \frac{1}{2\mathcal{D}_{in}(\mathcal{P})} \sum_{p_k \in \mathcal{P}} \mathcal{D}_{out}(p_k) \tag{17}$$

where $\mathcal{D}_{in}(\mathcal{P})$ is within class distance measured on whole set of partitions $\mathcal{P} = \bigcup_{n=1}^{N} p_n$.

The degree of grouping convey the information of how well the points X are unified together under the partitions while degree of separation tells us how well partitions separate the points X from each other. Ideally, we want degree of grouping be as small as possible or degree of separation close to one. Note that $G\{\mathcal{P}\} + S\{\mathcal{P}\} = 1$.

Then, the goodness of content-based classification method can be obtained using the *classification consistency* $\mathcal{F}\{\mathcal{P}\}$ defined on partition set $\mathcal{P}$:

$$\mathcal{F}\{\mathcal{P}\} \triangleq 10 \log_{10} \frac{S\{\mathcal{P}\}}{G\{\mathcal{P}\}} \tag{18}$$

We used the above classification consistency method (Equation 18) in evaluating the classification tree of the MPEG-2 encoded video stream. The effectiveness measurement can also help us to find the most important content features in determining the resource requirements.

3.3.2 Experimental derivation of MPEG-2 classification scheme

Using the classification consistency method, described in Section 3.3.1, our goal is to find an efficient classification tree for MPEG-2 video. We divide our approach, based on object-based scene classification scheme (Figure 7), into two parts. First, we choose a set of scene and object features for use in the classification scheme. Second, we construct and optimize a classification tree, representing efficiently the scene classification mechanism.

The selection of object features is based on measurements of the classification consistency at the first level of hierarchical scene segmentation scheme. We choose four features, having the highest value of classification consistency $\mathcal{F}$: *size*, *complexity*, *motion* and *camera operations*. As we will show later, these features have major influence on the resulting video source characteristics: both absolute value and shape of D-BIND traffic constrained function, selected as a traffic resource descriptor. According to Video Structure Model, described in Section 3.1, these features are assigned into the scene type and video object descriptors as follows. The scene type descriptor $STD(s_i)$

characterizes scene s_i according to three basic camera operations and the video object descriptor $VOD(o_j)$ describes video object o_j in the scene according to its size, speed, and complexity.

Based on our trace driven simulations, described in Section 5, we found that it is sufficient to quantize the scene and object visual features into three levels only. In that case, we approximate object size as 1/3, 2/3, and 3/3 of the whole image resulting in maximum of three objects in the scene. While the first layer of hierarchical scene segmentation scheme contains only one object of size 3/3 (whole image), the second layer can contain either two (of size 1/3 and 2/3) or three (of size 1/3) objects. Similarly, the object speed[‡] and complexity are also quantized into three levels (refer to Table 1).

camera operation	static	panning	zoom
object size	1/3	2/3	3/3
object complexity	smooth	medium	cluttered
object speed	low	medium	fast

Table 1: Scene and object features.

Note that although satisfactory video content analysis tools are available [13, 20], in this paper we focus on the study of the new approach of content-based traffic modeling. In this paper we use manual procedures to estimate several visual features of the video scene or objects, except that object/scene complexity is computed using the analytic formula in Equation 7. Other features, such as motion speed and camera operations, are roughly classified based on manual visual inspection. Our goal is to first study the "optimal" performance achievable by the content-based traffic models by isolating the possible errors made by the automatic video content analyzer. We will briefly address the impact of "imperfect" content analysis on the performance later in Section 5.

The scalability of the scene classification scheme results in a family of scene classification models, depicted in Figure 7. Depending on the scene segmentation layer and selected video features, the appropriate model consists of several classification stacks, representing the efficient scene classification scheme. For example, two layer scene segmentation scheme with three quantization levels, as we described earlier, results in the Model A. It includes three classification stacks A1, A2, and A3 corresponding to one, two, and three objects in the scene respectively. Note that the number of classes in Model A is relatively large (2457) and may not be practical for implementation. It is therefore desirable to decrease the number of classes by clustering and merging of the classes.

In our experiment, we found a satisfactory performance using the Model B, depicted in Figure 7. It consists of two stacks B1 and B2 for one and two objects respectively, resulting in total of 108 classes. Using the measure of classification consistency, Figures 8 and 9 depict the derivation of optimal classification tree of Stack B1 and B2 respectively. At each level the most efficient classifier, resulting in the best value of classification consistency $\mathcal{F}$, is selected. For example, at the first layer of Stack B1, three classifiers are compared: global complexity, global motion, and camera operations. The global complexity is selected at this level, because it has the highest value of classification consistency coefficient. The selection of subsequent coefficients at lower levels is similar. To further decrease the number of classes, we introduce the Model C resulting in 81 classes only. Figure 10 depicts the derivation of optimal classification tree of Model C. Note that in this specific example the combined classification tree in Figure 10 results in better classification consistency compared to Model B.

To illustrate the results of the content-based scene classification, in terms of class-to-resource

[‡]Note that the object speed pertains to the perceived motion rather than the "true motion". For example, in a scene where the camera is tracking a foreground object, the perceived speed of the foreground object is zero or slow.

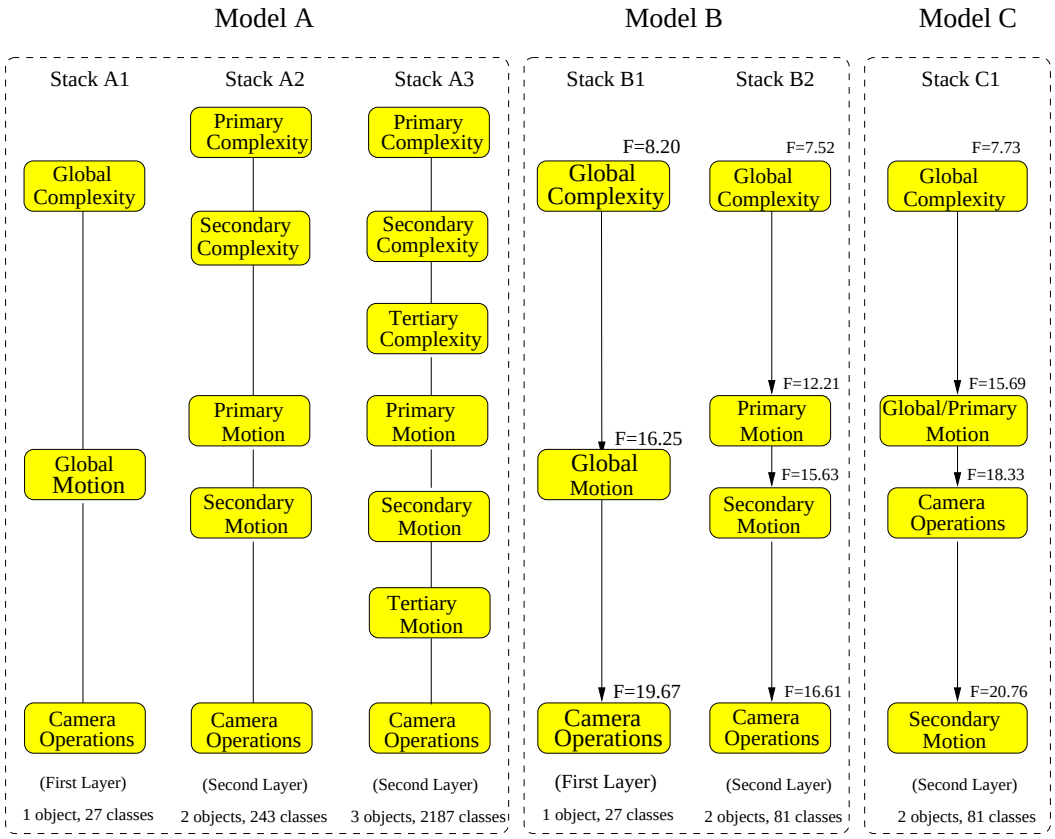

Figure 7: Family of scene classification models.

mapping, we show actual traffic constrained functions corresponding to the D-BIND traffic resource descriptors together with their standard deviation as obtained using the different classification schemes. Resource mapping based on camera operation and global motion is depicted on Figures 11 and 12 respectively. Inefficiency of the classification based on these features alone is demonstrated in very large values of standard deviation. Better results can be obtained using the complexity feature. Figure 13 depicts results obtained using the classification model based on complexity with two quantization levels. Even though only two quantization levels are used, the classification consistency is higher than the three level classification based on camera operations alone. This is also demonstrated in smaller values of standard deviation. The classification consistency can be significantly improved by adding more quantization levels to complexity, as shown in Figures 14 and 15. Figure 16 depicts a resource mapping based on two features: global complexity and global motion. The detailed classification results corresponding to smooth, medium and cluttered global complexity are depicted on Figure 17, Figure 18 and Figure 19 respectively. Finally, the resource mapping obtained by using three features (complexity, global motion, and camera motion) are depicted on Figure 20. From these results we can see how addition of more features improves the classification consistency. This is demonstrated by decrease of standard deviation of traffic constrained functions, corresponding to the appropriate class.

Table 2 shows an example of scalability of the scene classification based on the number of quantization levels of the complexity feature. We can see that the classification consistency is improving with the increasing number of quantization levels and asymptotically approaching the

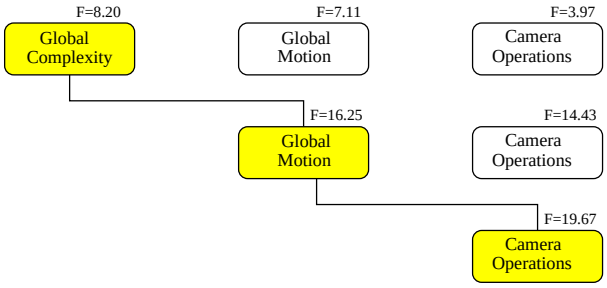

Figure 8: Experimental MPEG-2 content-based scene classification tree (First Layer, 1 object).

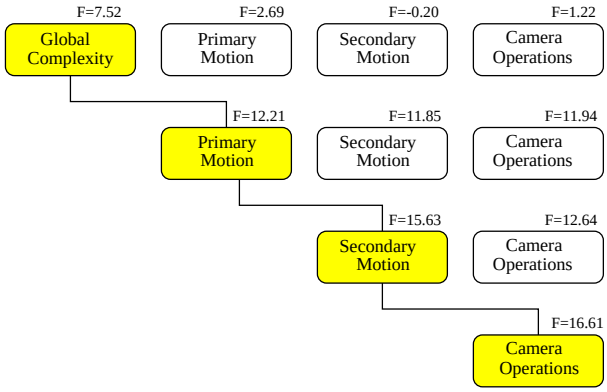

Figure 9: Experimental MPEG-2 content-based scene classification tree (Second layer, 2 objects).

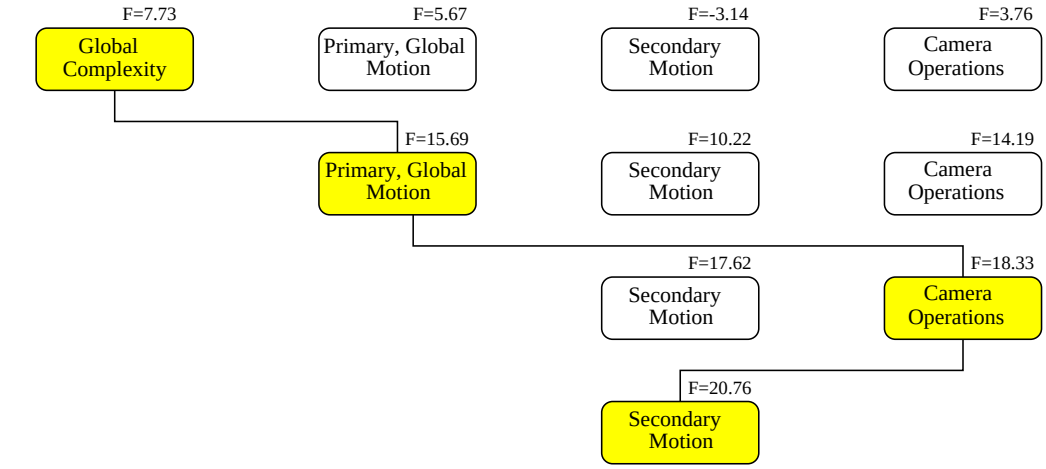

Figure 10: Experimental MPEG-2 content-based scene classification tree (Combined First and Second layer, 1-2 objects).

limiting value. In this specific example, the classification consistency is not substantially improved beyond quantization level 9 reaching $\mathcal{F} = 12.22$. At this point, the further improvement is possible only using more content features. For example, using only three quantization levels for both complexity and motion, the classification consistency improves to $\mathcal{F} = 15.69$. We can see, that this value is higher than using only single complexity feature quantized to ten levels, as shown in Table 2.

3.4 Content-based video segmentation

The ultimate goal of video segmentation algorithm is its efficient use in dynamic resource allocation leading to optimization of network resources and increase of network utilization.

Generally, we distinguish two video segmentation algorithms: *off-line* and *on-line* [5, 6, 7]. Off-line video segmentation algorithm can be applied to stored video streams; its main advantage is the possibility of obtaining the optimal renegotiation schedule. Off-line algorithm can lead to

Figure 16: Resource mapping based on complexity and global motion ($\mathcal{F} = 15.69$)

Figure 20: Resource mapping based on scene complexity, global motion and camera motion ($\mathcal{F} = 18.33$)

the optimal solution because the algorithm is not causal. In determination of the most effective video segmentation, the algorithm takes an advantage of knowledge of the full video trace history in advance. Since determination of the optimal renegotiation points is computationally extensive, heuristic off-line segmentation algorithms were also proposed [7]. These algorithms are typically less computationally extensive but lead to sub-optimal solutions only.

On-line video segmentation algorithms used for real-time traffic are causal: they are based on assumption of knowledge of present and previous trace history only. They incorporate heuristic traffic prediction models monitoring the incoming traffic, network queue length or cell loss to assess the future stream resource requirements. When future resource requirements exceed the current resource reservation, new resource reservation request is generated. On the other hand, when stream resource requirement is less than currently reserved, the request to decrease the resources may be generated. In general, effectiveness of on-line video segmentation algorithms depends on the accuracy of the traffic prediction. Since most of traffic prediction algorithms consider the past

Number of levels	F	Number of levels	F	Number of levels	F
2	3.97	5	11.72	8	12.21
3	8.20	6	11.98	9	12.22
4	9.97	7	12.05	10	12.22

Table 2: Relation of optimized classification consistency of global scene complexity on number of quantization levels

video trace history only, their accuracy is limited.

We propose a new video segmentation algorithm which allows more accurate traffic prediction. The *content-based video segmentation* algorithm is based directly on video content rather than bit-rate and it is closely related to the video scene classification, discussed in Section 3.3.

The content-based video segmentation divides video stream into renegotiation intervals based on scenes of different video content. The video content information can be extracted from the compressed video stream in real time [13] or supplied directly by a digital video camera and associated scripts. Future cameras might provide supplemental information about zooming speed, panning and other features related to the video content. The important feature of the content-based approach is its direct applicability to both off-line and on-line segmentation.

The content-based video segmentation has two phases. First, at the appropriate hierarchical scene segmentation layer, the frames and identified video objects are characterized in terms of scene type and video object descriptors. Then, each frame is classified into one of the predefined scene classes. The video segmentation is performed in the following way. The consecutive frames having the same content class are grouped together to form the scene of the particular class.

To summarize, the content-based segmentation can be expressed as follows. Denote video $V = \{f_k\}_{k=1}^{n}$ as a sequence of frames f_k. Denote scene s_i content class as C_i. Also, denote content-based classification of the frame as $f_j \rightarrow C_i$ and classification of the scene as $s_i \rightarrow C_i$. The goal of the content-based segmentation is to partition a video $V = \{s_i\}_{i=1}^{m}$ into a set of non-overlapping segments (scenes) $s_i = [f_k, f_{k+l}]$ of length l. Each segment consists of possibly a variable number of frames such that each frame belonging to the particular scene is classified into the same scene class (i.e. it has a similar content). We say that scene is classified into the class C_i when the following holds:

$$(s_i \rightarrow C_i) \equiv \{\forall f_j \in [f_k, f_{k+l}], f_j \rightarrow C_i\} \qquad C_i \in C \tag{19}$$

where C is a set of all scene classes. In other words, content-based segmentation divides the video frame sequence into segments, frames of which belong to the same scene class.

4 Content-based dynamic network resource allocation system

The content-based network resource allocation algorithm can be used in both real-time and non real-time systems. The main difference between both systems is that non real-time system has all the information about scene content, including the renegotiation points, scene length and resource requirements, available before the stream is sent to the network. Also, the non real-time content-based segmentation is more accurate, since it does not experience delay associated with the determination of the video content. In other words, the stream in segmented off-line. The real-time system is more complex, since it assumes no previous information about the stream. The real-time dynamic resource allocation systems can be used with video streams such as live news programs while non real-time content-based resource allocation can be used in video-on-demand applications.

The real-time dynamic resource allocation system is depicted on Figure 21. The system consists of single Link Resource Control (LRC) module and multiple Stream Resource Control (SRC) modules, one for each video stream.

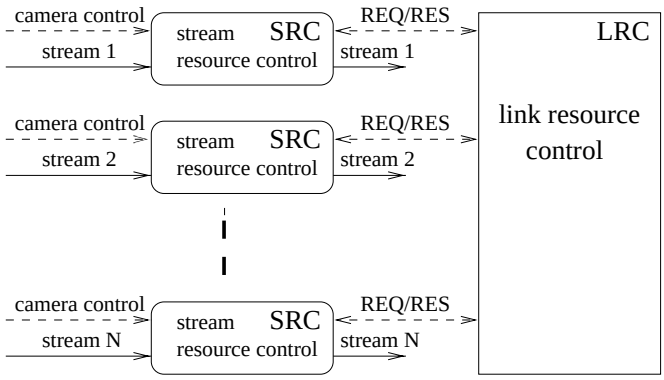

Figure 21: Real-time dynamic resource allocation system.

The LRC module contains algorithms for both admission control and real-time resource allocation functions. It is used by all streams sharing the link and is located at either the user network interface (UNI), the external ATM switch, or QoS enabled router. The LRC maintains the current stream resource reservation and QoS requirements. When the dynamic resource allocation request, generated by SRC, is received, it is checked against the available link resources. If there are enough resources available, they are allocated and reserved for the stream otherwise negative response is returned.

The SRC module manages the transport of each video stream to the network interface. By monitoring the incoming compressed video stream, it predicts its future network resource requirements. To maintain the QoS, SRC tries dynamically to renegotiate and modify currently held stream resources if they are not sufficient or if they exceed stream's future needs. In that case SRC sends a resource request message to the LRC. If resource request message to change the current reservation is not confirmed, to conform to its previously obtained resources the SRC module may continue to maintain the stream resource requirements by traffic shaping or if possible, by modification of the encoder parameters (such as the quantization parameter in MPEG-2). Otherwise, when new resources are successfully obtained, the stream is sent to the network interface without any changes.

The real-time content-based SRC module is in more detail depicted on Figure 22. It contains following five modules: Content Analyzer/Classifier, Traffic Analyzer, Class Resource Prediction, Resource Control, and Resource Shaping.

The input compressed VBR traffic stream, generated by the VBR encoder, is analyzed by Content Analyzer/Classifier. The scene visual content is extracted using an automated analysis of compressed video stream or supplied externally by a digital video camera. Additional external information, which can be used by the content analyzer, is the information about scene cuts, storyboards describing scene activities, and internal camera operations. Fully automated analysis of compressed video signals has shown great promise in past few years [13] and can provide at least satisfactory approximation for the purpose of content-based traffic modeling. The Content Analyzer/Classifier segments video stream into scenes described by content descriptors and classify them into corresponding scene classes. Once a video scene is mapped to a scene class, the repre-

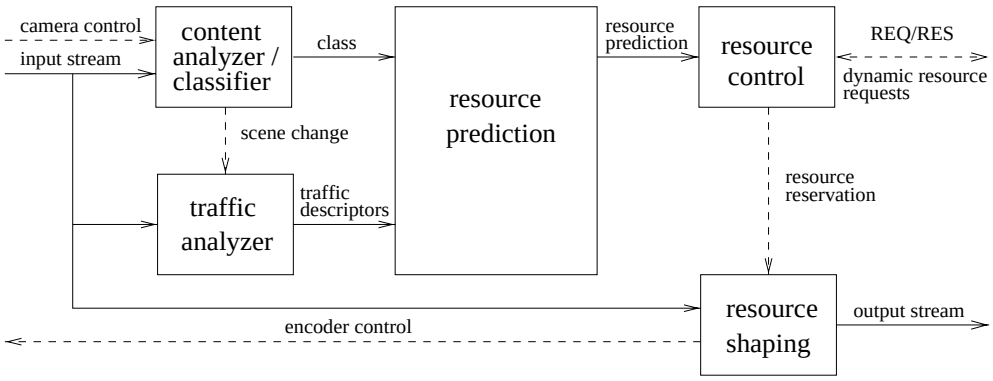

Figure 22: Stream Resource Control (SRC) module.

sentative traffic descriptor for that particular class can be used to determine resource requirements valid until the next scene change. New request for the resource increase/decrease may be launched at the beginning of next scene. Depending on the compression algorithm, the determination of the scene content in real-time may occur a delay of several frames (such as in MPEG-2). Our simulation results, described in Section 5, will show that delay of 12 or 24 frames in scene analysis/classification does not significantly decrease the network utilization.

The Traffic Analyzer uses information about the scene cuts such that each time a new scene starts, the D-BIND traffic resource descriptor is evaluated starting from the first frame of the new scene. Its evaluation then continues with each new frame until the end of the scene. At the end of each scene, D-BIND traffic descriptors are sent to the Class Resource Prediction module to be stored at the internal class cache and to possibly update the representative scene descriptors of the corresponding class.

The Class Resource Prediction module maps scene class to its corresponding resource requirements. The representative resource requirements for each class used for the prediction are updated at the end of each scene. Also, at the end of each scene, these traffic descriptor parameters are stored at the internal class cache. The cache keeps the most recent scene classes and corresponding traffic resource descriptors for the future prediction.

The caching scheme results in better prediction of the future resource requirements. The advantage which can be obtained by the use of the cache can be explained by the common structure of the video. For example, assume a video showing the dialog between two persons inside the room; such movie can be edited in such a way that views of both speakers alternate from scene to scene. Each scene is classified by content classifier, its traffic resource is determined by traffic analyzer and stored in the cache. Without loss of generality we assume that traffic descriptors corresponding to the same camera view and visual features (e.g. motion speed) have similar resource requirements. If the current scene class is found at the cache, its corresponding resource requirements are used as predictor for the current scene. If cache does not contain the resource requirements of the current scene class, the representative resource requirements for this class are used instead.

The Resource Control utilizes information about resources corresponding to the current scene and allocated network resources in deciding a new renegotiation request. If required network resources cannot be obtained, the traffic stream must be modified at the Resource Shaping module which controls output to the network buffer.

The Resource Shaping module polices and shapes the video stream such that it conforms to the

currently negotiated resource parameters. The simple policing of the D-BIND constrained sources can be accomplished by a set of leaky buckets [6]. In some cases video encoder parameters can also be dynamically changed such that video stream complies with the currently reserved network resources.

5 Trace-driven simulation

Figure 23 depicts the model of an entry node ATM multiplexer with dynamic resource allocation which we have used to design our trace driven simulator for our real-time content-based (CB-rt) and our non real-time content-based (CB-nrt) schemes. For performance comparison, we used our trace driven simulator also for the real-time renegotiated VBR scheme (RVBR-rt) and non real-time renegotiated VBR (RVBR-nrt) schemes, both proposed in [6]. We assume N video streams multiplexed over the single communication line or ATM virtual path (VP) of bandwidth $c = 45Mbps$ using a FCFS network scheduling policy. Each source in the simulator is based on 54000-frame-long trace (0.5 hour) of MPEG-2 encoded movie *Forrest Gump*. Each source starts at a random frame within the trace and wraps around to the start when end of trace is reached. To accelerate the simulation, each source also contains information about renegotiation points, which were pre-computed using appropriate on-line or off-line video segmentation algorithms. At the renegotiation point, the source initiates resource renegotiation request. The Link Resource Control module then decides, based on currently available resources, to either accept or reject this request. For simplicity, we assumed that once the request for more resources is rejected, the source is blocked and the new request is generated at the next renegotiation point only. Other renegotiation policies suggest that source, once rejected, repeatedly try to renegotiate at the later time or source can apply traffic shaping to accommodate the stream into the currently available resources. These other options have not been incorporated in this paper. Video streams are buffered at the network buffer from which they are transmitted to the network. The network buffer occupancy is used by the D-BIND dynamic resource allocation algorithm, as described in the next paragraph.

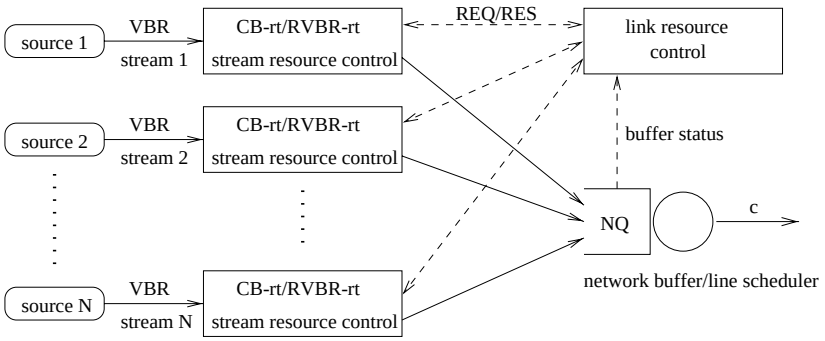

Figure 23: Trace-driven simulation model for CB-rt (content-based) and RVBR-rt dynamic resource allocation.

The Link Resource Control module contains real-time dynamic resource allocation control algorithm for D-BIND constrained sources s_i [6] defined by rate-interval pairs $R_T^{(i)} = \{(r_k^{(i)}, t_k) \mid k = 1, 2, \cdots, P\}$. Denote Q the network buffer size and $Q\tau$ network buffer requirements at time τ when the new renegotiation request arrives. Also define subset $\mathcal{A}_\tau = \{s_i \mid i = 1, \cdots, n\}$ of n sources s_i currently accepted by the link resource control module at time τ while the source s_{n+1} is initiating

the renegotiation request.

For the FCFS scheduling policy the required network buffer size Q_τ is:

$$Q_\tau = \max\{0, \max_k\{t_k(\sum_{i=1}^{N} 1_{\{s_i \in \mathcal{A}\}} r_k^{(i)} + r_k^{(n+1)} - c)\}\} \qquad k = 1, \cdots, P \tag{20}$$

where N is a number of sources and c is the link speed. The real-time admission algorithm is based on the observation that the dynamic renegotiation control has to take into account the network buffer occupancy. Denote a O_τ buffer occupancy at time τ. Then, any renegotiation can be considered only if currently available buffer space is greater than or equal to total buffering requirements of all sources:

$$R_\tau \leq Q - O_\tau \tag{21}$$

We have used Monte Carlo trace-driven simulation in the study of performance of the different on-line and off-line segmentation schemes. In particular, we compared CB-rt and CB-nrt content-based segmentation to renegotiated VBR (RVBR-rt, RVBR-nrt), and renegotiated CBR (RCBR-rt) segmentation [5, 6] . For a given renegotiation blocking probability, we obtained the maximum number of streams, multiplexed on the line, for a shared network buffer size equivalent to delay of 0 to 0.30 s. The trace driven simulator uses the following algorithm:

```
1.  Fix buffer size and initial number of multiplexed sources (default 1)
2.  Run simulation 100 times and calculate rejection probability
3.  If rejection probability is less than the maximum specified,
      increase number of sources and run simulation again from 2
4.  Else save results (number of sources)
5.  If needed,
      select another buffer size and start from 1
6.  Else end simulation
```

In the first experiment we studied the effectiveness of the off-line content-based renegotiation algorithms. We run three simulations to compare performance of our off-line content-based renegotiation algorithm to the most effective frame-based segmentation and RVBR-nrt [6] off-line segmentation algorithms.

First simulation of the frame-based segmentation algorithm was used as an indicator of the performance upper bound. In the frame-based renegotiation scheme the resources (equivalent to the number of bits in the frame) are requested every frame thus creating the bottleneck in the dynamic link resource control module due to the very large number of renegotiation requests. The high link utilization of the frame-based segmentation is due to its ability to extract statistical multiplexing gain of both short as well as long time scale variations of the video source. The second simulation investigates the effectiveness of the off-line content-based renegotiation algorithm. The video trace was segmented according to the content-based segmentation algorithm and for each scene the exact D-BIND traffic resource descriptors were obtained. The mean interval between renegotiations for the content-based video segmentation was 3.3 s. The third simulation used RVBR-nrt off-line renegotiation algorithm [6]. We adjusted its single parameter ψ ($0 \leq \psi \leq 1$) which controls the renegotiation frequency such that the renegotiation frequency was about the same as that of content-based approach. We used $\psi = 0.65$ resulting in a mean interval between renegotiations 2.58 s, slightly less than that in content-based approach.

Figure 24 depicts link utilization corresponding to three off-line video segmentation algorithms for the stream renegotiation blocking probability of 10^{-2}. We can observe the utilization of the

Figure 25: Effectiveness of the on-line content-based CB-rt and RVBR-rt segmentation algorithms.

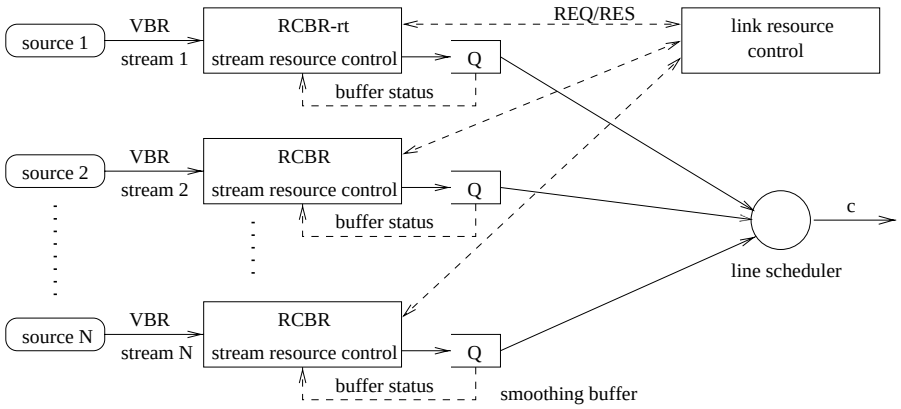

Figure 26: Trace-driven simulation model for RCBR-rt dynamic resource allocation.

frame-based renegotiation scheme ranges from 85% (without network buffer) to 95% (with a buffer size equivalent to 0.25 s delay). While the utilizations of both CB-nrt content-based and RVBR-nrt renegotiations are only 31 % for the case without network buffer, their utilizations gradually increase with the size of the network buffer and they approach the bounding link utilization corresponding to the frame-based scheme. The off-line content-based scheme shows about 5% improvement in the link utilization compared to the RVBR-nrt scheme for network buffer delay of 0.25 s.

In the second experiment we were interested in the effectiveness of different on-line renegotiation algorithms. Figure 25 depicts results of five different simulations. Two CB-rt curves correspond to simulations of real-time content-based segmentation. Since resource requirements in the real-time content-based algorithm are predicted from the scene class, we run two different simulations accounting for various delays caused by the on-line video content analyzer. First, we assumed an ideal content classifier, which recognizes the current scene class with zero delay. Second, we assumed 24 frame delay in scene classification. Our results show that when the content classifier incurs a 24 frame delay, the performance of dynamic allocation scheme slightly decreases for about 5% for a large network buffer of 0.25 s. In the case of small network buffers, the delay caused by on-line scene classification does not have any substantial effect on the performance. Note that the content-based approach significantly outperforms the RVBR-rt approach despite that a lower renegotiation frequency is used for the content-based approach.

For comparison we also run simulations with segmentation obtained from RVBR-rt online algorithm; results are also depicted in Figure 25. In the RVBR-rt on-line renegotiation scheme parameters α and β control the renegotiation rate [6]. We selected different values of parameters: ($\alpha = 1.1$, $\beta = 0.9$), ($\alpha = 1.2$, $\beta = 0.8$), and ($\alpha = 1.3$, $\beta = 0.7$) resulting in 1.15, 2.23, and 4.23 s/request mean interval between renegotiations respectively. The link utilization for the RVBR-rt renegotiation was substantially less than those of using the content-based approach (about 55% - 70% difference). Also, we can see that the renegotiation frequency in RVBR-rt algorithm has a significant influence on the link utilization.

In the third experiment we used simulation model of the RCBR-rt real-time renegotiation scheme [5] depicted in Figure 26. Similarly, N video streams are multiplexed over the single communication line or ATM virtual path (VP) of bandwidth $c = 45Mbps$ using FCFS network scheduler policy. But the RCBR-rt segmentation algorithm uses separate buffers for each stream and no shared network buffer [5]. Its on-line bandwidth prediction algorithm is based on an AR(1)

Figure 27: Effectiveness of the on-line content-based CB-rt, RVBR-rt, and RCBR-rt algorithms

Figure 27 compares simulation results of real-time CB-rt (content-based), RVBR-rt and RCBR-rt segmentation algorithms showing achieved utilization depending on required buffer per stream. The best performance is achieved by the real-time content-based segmentation: its network utilization sharply increases from 31% for no network buffer up to 95% for the buffer of the size 20 kbytes per stream. Performance of RVBR-rt algorithm depends on renegotiation frequency. For the network buffer size of 20 kbytes its link utilization is only 67% even for a high renegotiation rate (1.2 s). Because of its separate buffering, the RCBR-rt segmentation algorithm has very low utilization at the small buffer size per stream, but its utilization increases sharply at the buffer size of 20 kbytes reaching 75% utilization for the 1.15 s average renegotiation rate. Its performance has a high dependence on the renegotiation frequency. The size of the required separate network buffer increases with the decrease in renegotiation frequency. For example, to reach the same utilization of 75% for a renegotiation rate of 2 s/request, each stream's separate network buffer must be three times larger than that with a 1.15 s/request renegotiation rate.

6 Conclusion

In this paper we have presented a new content based approach to the modeling of VBR resource requirements. This approach is suitable for resource prediction in dynamic resource allocation

schemes of VBR video over links in which QoS cannot be easily guaranteed. In bandwidth limited networks (such as ATM interface or wireless), dynamic resource allocation can substantially increase the link utilization and also decrease the required network buffering. This is possible by allowing each single stream to request resources in the real time on the need base instead of reserving them only once at the beginning of the session. While the static resource allocation scheme leads to over-reservation of resources, the dynamic bandwidth allocation is able to extract the burst level multiplexing gain of the VBR video while providing the QoS protection.

The effectiveness of the real-time dynamic resource allocation depends on prediction of the stream resource requirements. To increase the accuracy of resource prediction, we proposed a model exploring the visual content of the video, an important indicator of the VBR stream bandwidth requirements. In our simulation experiments we used the actual MPEG-2 VBR trace for which we identified the major video content features and proposed the simple, but effective video content classification scheme. We also show how the visual content, when extracted and classified, can be directly mapped into the stream resource requirements.

We used the trace-driven simulator to compare on-line and off-line content-based dynamic renegotiation (CB-rt and CB-nrt) with several other renegotiation algorithms. In particular we performed simulations of frame-based, RVBR, and RCBR algorithms. While the frame-based segmentation sets an upper bound of performance because if its ability to extract the short and long time scale multiplexing gain, it creates very high volume of renegotiation requests. Therefore it is not of practical interest. On the other hand, with 3.3 s average renegotiation period the off-line content-based video segmentation approaches the performance bound for network buffer size equivalent to 0.3 s (assuming link speed c=45 Mbps). In the simulation of off-line algorithms we found, using 0.5 hour actual video trace, that our CB-nrt algorithm outperforms RVBR-nrt segmentation by about 5% for buffer size of 0.3 s. While this result might vary with different video, the advantage of the content-based approach is in its relatively fast speed (compared to other video segmentation algorithm). For the real-time case, our simulations show that our CB-rt algorithm with 3.3 s average renegotiation period approaches 95% utilization for large buffers (0.3 s), the RVBR-rt approaches only 68% utilization for much higher renegotiation frequency (1.15 s average renegotiation period) and the same buffer size. Because of its separate buffering, the RCBR-rt segmentation algorithm has very low utilization at the small buffer size per stream and its performance has a high dependence on the renegotiation frequency. From our simulations we conclude that while the performance of off-line content-based segmentation shows only marginal improvement compared to RVBR-nrt algorithm, its on-line performance substantially exceeds both the RVBR-rt and RCBR-rt algorithms.

References

[1] H. Zhang and D. Ferrari, *Improving Utilization for Deterministic Service In Multimedia Communication,* Proceedings of IEEE International Conference on Multimedia Computing and Systems, 1994.

[2] K. Liu, D. W. Petr, V. S. Frost, H. Zhu, C. Brown, and W. Edwards, *Design and Analysis of a Bandwidth Management Framework for ATM-Based Broadband ISDN,* IEEE Communications Magazine, May 1997, pp. 138-145.

[3] J. Y. Hui, *Resource Allocation for Broadband Networks*, IEEE Journal on Selected Areas in Communications, Vol. 6, No. 9, December 1988, pp. 1598-1608.

[4] A. Adas, *Supporting Real Time VBR Video Using Dynamic Reservation Based on Linear Prediction,* Proceedings of IEEE INFOCOMM'96, pp. 1476-1483.

[5] M. Grossglauser, S. Keshav, and D. Tse, *RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic,* Proceedings of SIGCOMM'95, September 1995, pp. 219-230.

[6] H. Zhang and E. W. Knightly, *A new approach to support VBR video in packet-switching networks*, Proceedings of NOSSDAV'95, April 1995, pp. 275-286.

[7] S. Gumbrich, H. Emgrunt, and T. Brown, *Dynamic bandwidth Allocation for Stored VBR Video in ATM End Systems*, Proceedings of IFIP'97, April 28 - May 2, 1997, pp. 297-317.

[8] V. S. Frost and B. Melamed, *Traffic Modeling For Telecommunications Networks*, IEEE Communications Magazine, March 1994, pp. 70-81.

[9] D. N. Tse, R. G. Gallager, and J. N. Tsitsiklis, *Statistical Multiplexing of Multiple Time-Scale Markov Streams,* IEEE Journal on Selected Areas in Communications, Vol. 13, No. 6, August 1995, pp. 1028-1038.

[10] E. W. Knightly and H. Zhang, *Traffic Characterization and Switch Utilization using a Deterministic Bounding Interval Dependent Traffic Model,* Proceedings of IEEE INFOCOMM'95, April 1995, pp. 1137-1145.

[11] P. Bocheck and S.-F. Chang, *A Content Based Video Traffic Model Using Camera Operations,* Proceedings of ICIP'96, September 1996.

[12] H. Sanderson and G. Crebbin, *Image segmentation for compression of images and image sequences,* IEE Proceedings: Visual Image Signal Processing, Vol. 142, No. 1, February 1995.

[13] J. Meng and S.-F. Chang, *Tools for Compressed-Domain Video Indexing and Editing,* Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Database, Vol. 2670, San Jose, February 1996.

[14] J.-P. Leduc, P. Delogne, *Statistics for variable bit-rate digital television sources*, Signal Processing: Image communication 8 (1996).

[15] S. Siggelkow, R.-R. Grigat, A. Ibenthal *Segmentation of Image Sequences for Object Oriented Coding*, Proceedings of ICIP'96, September 1996.

[16] A. N. Netravali and B. G. Haskell *Digital Pictures: Representation and Compression*, Plenum Press, New York.

[17] L. Torres and M. Kunt, eds. *Video Coding: The 2nd Generation Approach*, Kluwer Academic, Boston, Massachusetts, 1996.

[18] A. Murat Tekalp *Digital Video Processing*, Prentice Hall, 1995, ISBN 013-190075-7.

[19] S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong *VideoQ- An Automatic Content-Based Video Search System Using Visual Cues*, ACM Multimedia Conference, Nov. 1997, Seattle, WA, also Columbia University/CTR Technical Report, CTR-TR #478-97-12. (demo: http://www.ctr.columbia.edu/videoq).

[20] J. Meng and S.-F. Chang, /it CVEPS: A Compressed Video Editing and Parsing System, ACM Multimedia Conference, Boston, MA, Nov. 1996.