Prioritized MPEG-4 Audio-Visual Objects Streaming over the DiffServ

HUANG Tian-yun¹, ZHENG Chan²
(1. College of Computer Science and Technology Southwest University for Nationalty Chengdu 610041 China; 2. College of Information, South China Agriculture University Guangzhou 510642 China)

Abstract The object-based scalable coding in MPEG-4 is investigated, and a prioritized transmission scheme of MPEG-4 audio-visual objects (AVOs) over the DiffServ network with the QoS guarantee is proposed. MPEG-4 AVOs are extracted and classified into different groups according to their priority values and scalable layers (visual importance). These priority values are mapped to the IP DiffServ per hop behaviors (PHB). This scheme can selectively discard packets with low importance, in order to avoid the network congestion. Simulation results show that the quality of received video can gracefully adapt to network state, as compared with the 'best-effort' manner. Also, by allowing the content provider to define prioritization of each audio-visual object, the adaptive transmission of object-based scalable video can be customized based on the content.

Key words video streaming; quality of service (QoS); MPEG-4; audio-visual objects (AVOs); DiffServ; prioritization

The transmission (or streaming) of the video applications over the ‘best-effort’ Internet is still a challenge today. However, as the progress of video compression and network technology, the demand for networked video applications is likely to increase significantly in the next few years. Researchers all over the world are dedicated to every aspects of video streaming in order to provide the quality of service (QoS) guarantees. Devotions in video coding technology include source coding algorithms such as scalable coding (spatial, temporal, SNR), MPEG-4 fine granularity scalability (MPEG-4 FGS), etc.; joint source/channel coding for error control and error resilience such as Reed-Solomon FEC codes, data partitioning, data recovery (RVLC), etc., and error concealment is also used at the receivers to get a better presentation quality[1-4]. Meanwhile, many end-to-end adaptive control strategies based on RTP/RTCP, are also proposed and improved to some exciting degree[5].

Besides these efforts, the proliferations of the MPEG-4 standard have attracted the eyes of many researchers[6]. MPEG-4 is a digital multimedia protocol for representing, manipulating and transporting natural and synthetic multimedia contents, such as audio, visual and data over a broad range of infrastructures. MPEG-4 provides an integrated object-oriented representation of multimedia content for the support of new ways of communication, access, and interaction with digital audio-visual data. MPEG-4 allows encoding of different audio-visual objects (AVOs) in a scene independently, each objects may require different QoS. To achieve a better transmission of MPEG-4 scene, classifying these objects into different priorities and mapping these priorities to the MPEG-4 transport layer are needed.

While the DiffServ gives the differentiated treatments on traffic aggregates, as compared with the ‘best-effort’ Internet nowadays[7]. There is little detailed work on the prioritized MPEG-4 AVOs transmission over the DiffServ. Most of these works, described in Refs.[7-9], are concentrated on MPEG1/2 and H.261/263 coded video. The quality adaptation mechanisms of streaming video are always based on IP packets scheduling, not on AVOs. A simple priority data-dropping idea is proposed in Ref.[7]. An automatic translation from adaptation policies to priority assignments on the basic units of video data, i.e., from application data units (ADUs) to streaming data units (SDUs) is also suggested. The dropping policy is based on priority mapping of ADUs and TCP Window scaling, only 2 types of dropping, frame or spatial, are allowed. The detailed QoS mapping of the packet video in the DiffServ is investigated in Ref.[8], where a video categorization based on relative priority index (RPI) and an adaptive QoS mapping mechanism of video packets from RPI onto different DiffServ levels is proposed and evaluated on ITU-T H.263+ coded video[9]. Recent work on AVOs streaming over
IP network is addressed in Ref.[10], where MPEG-4 AVOs are dynamically added or dropped based on their QoS parameters, in order to conform to the TCP friendly rate control (TFRC) mechanism. However, no explicit algorithm of QoS parameters mapping to priority is suggested, and the priority mapping to the DiffServ per hop behaviors (PHB), is not mentioned.

In this paper, we propose a priority-mapping scheme for MPEG-4 AVOs, and give the architecture for prioritized AVOs streaming over the DiffServ. First, AVOs in the scene are extracted and prioritized according to their visual importance; these priorities are mapped to the DiffServ PHB for selectively discarding. AVOs are then compressed into elementary stream, partitioned and packetized for transmission. A significant QoS improvement of streaming video can be provided by this scheme.

1 Scalable Video Encoding

Scalable encoding — also called hierarchical or layered encoding — is an encoding technique that is particularly well suitable for networked video applications. It appears first in the MPEG-2 standard, and later in H.263+ (enhanced version of H.263) and MPEG-4[2,9]. It was proposed to increase the robustness of video codec against network packet loss[11]. The main concept of scalable encoding is to encode the video into several complementary layers: the base layer (BL), and one or several enhancement layers (ELs). The base layer is a low quality version of the video, which must be decoded in order to get a minimum acceptable video quality. Decoding each enhancement layer successively will progressively enhance the rendering quality of the video. Obviously, in order to decode the enhancement layer $n$, the decoder need all lower order layers, from base layer to layer $n-1$.

However, the scalability is based on ordinarily rectangular frame, and only free in two degrees, namely, spatial and temporal dimensions. Spatial scalability involves an improvement in the resolution of the video frame sample, whereas temporal scalability involves changing the frame rate during play out. Other scalability techniques also exist, but they tend to be fundamentally based on these two dimensions, for example, SNR, or multiple description encoding (MDE).

While exploiting MPEG-4 AV objects capability, a more freedom of scalability can be implemented based on AVO concept, which allows us to include or omit individual object or only part of its enhancement layers according to network condition. Scalability of MPEG-4 is also much more versatile than simple frame based scaling, because each object in the scene can be adapted or modified independently. On the contrary, frame based scalability may affect the whole of the scene; while in the case of object based scalability, only the objects are degraded independently, so a better perceptual quality can be provided. A conceptual illustration of MPEG-4 object-oriented encoding is given in Fig.1.

![Fig.1 Example of object-based scalability](image1)

2 Object Description in MPEG-4

Object-oriented representation of multimedia content is the most salient characteristic in MPEG-4. Every multimedia element to be compressed is represented as individual AVO. The combination of these elements is assured by the scene description language – binary format scene (BIFS). The delivery of the media is defined in the delivery multimedia integrated framework (DMIF). MPEG-4 architecture includes three layers: the compression layer, the sync layer, the delivery layer, as shown in Fig.2.

![Fig.2 MPEG-4 architecture (@MPEG-4)](image2)
The compression layer generates representation of content data called elementary streams (ESs). ESs are the basic abstraction for any data source. The hierarchical relations, positions and properties of ESs are described by dynamic set of object descriptors (ODs). ODs may be conveyed through one or more ESs. ESs are partitioned into access units (AUs), which are conveyed into a sequence of packets called SL-PDUs on the sync layer. The SL-PDUs are subsequently transmitted to the delivery layer for multiplexing and generating a FlexMux stream.

The purpose of the object descriptor framework is to identify, describe and associate elementary streams with the various components of an audio-visual scene. A media object is associated to its elementary stream resources via an object descriptor.

An object descriptor is a collection of one or more ES descriptors that provide configuration and other information for the streams that relate to a single object (media object or scene description). Each object descriptor is assigned a unique identifying number (object descriptor ID). This identifier is used to associate media objects in the scene description with a particular object descriptor, and thus the ESs relate to that particular object. MPEG-4 has defined many objects descriptors for different purpose: scene descriptor, ES descriptor, IPI descriptor, and QoS descriptor, etc.

<table>
<thead>
<tr>
<th>QoS Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX_DELAY</td>
<td>Absolute maximum end to end delay for the stream</td>
</tr>
<tr>
<td>PREF_MAX_DELAY</td>
<td>Preferred maximum end to end delay for the stream</td>
</tr>
<tr>
<td>LOSS_PROB</td>
<td>Allowable loss probability of any single AU</td>
</tr>
<tr>
<td>MAX_GAPLOSS</td>
<td>Maximum allowable number of consecutively lost AUs</td>
</tr>
<tr>
<td>MAX_AU_SIZE</td>
<td>Maximum size of an AU</td>
</tr>
<tr>
<td>AVG_AU_SIZE</td>
<td>Average size of an AU</td>
</tr>
<tr>
<td>MAX_AU_RATE</td>
<td>Maximum arrival rate of AUs</td>
</tr>
<tr>
<td>PRIORITY</td>
<td>Priority for the stream</td>
</tr>
</tbody>
</table>

QoS descriptor is the most important descriptor to support video transmission over the QoS-enabled network, which can be used for classification of media objects. QoS descriptor conveys the requirements that the ES has on the transport channel and a description of the traffic that the ES will generate. Tab.1 shows QoS parameters used in the OD of an ES. In our work, we only take the ‘Priority’ metric into consideration, which is used to mark different objects and their layers for prioritized packets discarding policy. Extensive work is going on in order to get a deep insight into this field.

3 MPEG-4 AVOs Classification

In this section, we will demonstrate that how to construct the priority structure for different media objects, i.e. classification and assignment of priorities to AVOs based on their visual importance. The AVOs are first extracted from the scene by video application layer. The typical scenario includes one or more people in the foreground and a background, as shown in Fig.3.

All extracted AVOs are given the initial priorities \( \{P(O_d)\} \). Assume that each AVO has multiple layers \( \{M_d\} \), each layer of the AVO has priority \( \{L_d\} \). The purpose of the object-priority assignment algorithm (OPA) is to calculate the priority values of each layer in each AVO, according to the initial priority marked by video application layer, and constructs packets based on priority values. Note that the selection of AVOs and the initial priority of each object are application-dependent, it is also constrained by different layered object encoding algorithms. As defined in MPEG-4, the visual importance of different objects in the scene may be marked by the author or content provider. Three persons are extracted from the scene in Fig.3, and marked with initial priorities (Maybe there exist objects with the same initial...
priority). Obviously, person O2 has the highest priority (P(O2)=0), whereas the background has a lower priority (P(O1)=2). (As in 8-priority level, value 0 is the highest priority, and value 7 is the lowest priority, in descendent order). These initially extracted AVs are the entire base layers with higher priorities; Priorities of all other enhancement layers are calculated according to the base layers, and always lower than the base layers,

Suppose 8 priorities are used in our example (16 or more priority levels can be used, with a higher complexity). The algorithm first determines the scope of those initially extracted AV objects — [P_m1, P_m2] (in Fig.3, P_m1=0, P_m2=2); all base layer has the same priority value equals to its object (L_k=P(O_k)). Second, the algorithm tries to find the enhancement layers of the AVs and gives them corresponding priority values, i.e., base layers are more important than enhancement layers. While network congestion occurred or there is only limited bandwidth for the video streaming, we protect base layer of AVOs in order to provide an acceptable video quality, parts of the enhancement layers of AVOs won’t be transmitted. In the worst case, we only transmit those base layers with relatively higher priorities; base layers with high priorities are discarded only if the network is seriously congested. Now, the graceful quality degradation can be guaranteed by this policy.

We briefly describe the algorithm and list its code below. The algorithm first receives AV layers and initial priorities from video application layer, sets the priority value for the base layer, and finds the priority scope of AVs. Second, check the layer structure of all AVs in the priority scope as follows: 1) If the enhancement layer is the second layer, its priority value is increased by 1 from the maximal value that has been assigned, which promise that the enhancement layers of the AVs with highest priority are transmitted first. 2) For other layers, the priority value is still increased by 1, but based on all previous enhancement layers. All priority values that have been assigned are recorded for next iteration.

1. function A = OPAA(P, M, Pmin, Pmax)
2. P_m = Pmax; P_m2 = Pmax;
3. N = length(P);
4. L = ones(N, max(M)); % -1: no such layer exists for current object;
5. for k=1:N, L(k, 1) = min(P(k), Pmax); end % initial priorities;
6. Ps = min(P); P_m = max(P);
7. Ps = max(P);
8. for p=P_m: P_m2
9. for k=1:N
10. q = M(k); s = 2;
11. while(P(k) = p) & (q=1)
12. if (s = 2) L(k, s) = min(Ps + 1, Pmax);
13. else L(k, s) = min(L(k, s-1), Pmax); end
14. if (L(k, s) > Ps) Ps = L(k, s); end
15. s = s + 1; q = q-1;
16. end
17. end
18. A = L_k

Fig.3 is an illustration of the OP algorithm for AVs, where P=[2,0,1,2], M=[1,3,2,2], P is the initial priority matrix and M is the matrix of the object’s layers. A=OPAA(P), M, 0,7), the result of A=[2, -1, -1, 0, 3, 4, 1, 5, -1; 2, 6, -1], where A_j is the layer j priority value of AVO. –1 means that current AVO has no such layer, so as to avoid confusion with the highest priority (here it is 0). The output matrix A is object priority matrix (OPM). Now, the priority information of all AVOS in a scene can be stored in a simple OPM matrix for lower system level packetization, if the content provider has initially marked the AVOS.

Fig.4 shows a visual demonstration of server packetizing and scheduling sequence. Obviously, the sequence should be first from left to right, then bottom to top, according to packet priority and network condition. Some less important packets may be discarded during congestion. We only use one queue arranged in descendent priority order for server scheduling. More complex scheduler may take group of VOPs (GoV) pattern into consideration, in which packets in P-VOP are less important than packets in I-VOP, and packets in B-VOP are less important than packets in P-VOP. Multiple priority queues may be required in this situation.
4 Priority-Mapping to DiffServ PHB

The differentiated service framework enables QoS provisioning within a network domain by applying rules at the edges to create traffic aggregates and coupling each of these with a specific forwarding path treatment in the domain through use of a CodePoint in the IP header\cite{6,12}. Packets entering a DiffServ domain (DS) are classified and traffic conditioned by a value called differentiated services CodePoint (DSCP). The CodePoint space is divided into three pools for the purpose of CodePoint assignment and management as shown in Tab.2\cite{12}.

<table>
<thead>
<tr>
<th>Pool</th>
<th>CodePoint</th>
<th>Assignment Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XXXXX00</td>
<td>Standards action (EF, AFxy, Default) class-Selector Code Points</td>
</tr>
<tr>
<td>2</td>
<td>XXXX11</td>
<td>Experimental/Local usage</td>
</tr>
<tr>
<td>3</td>
<td>XXXX01</td>
<td>Experimental/Local usage/Future standards</td>
</tr>
</tbody>
</table>

The recommended CodePoint 000000 is used for best effort forwarding behavior for compatibility. The class selector PHBs XXX000 are reserved to ensure that DS-compliant nodes can coexist with IP precedence-based nodes. The expedited forwarding (EF) PHB is intended to provide a building block for low-delay, low-jitter and low-loss services by ensuring that the EF aggregate is served at a certain configured rate, whereas the assured forwarding (AF) PHB tries to offer different levels of forwarding assurances for IP packets crossing a DS domain\cite{13-14}. The class/precedence of AF group is given in Tab.3.

<table>
<thead>
<tr>
<th>Drop Precedence</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>AF11 – 001010</td>
<td>AF21 – 010010</td>
<td>AF31 – 01010</td>
<td>AF41 – 10010</td>
</tr>
<tr>
<td>Medium</td>
<td>AF12 – 001100</td>
<td>AF22 – 010100</td>
<td>AF32 – 01100</td>
<td>AF42 – 100110</td>
</tr>
<tr>
<td>High</td>
<td>AF13 – 001110</td>
<td>AF23 – 010110</td>
<td>AF33 – 01110</td>
<td>AF43 – 100110</td>
</tr>
</tbody>
</table>

As seen from the OPA algorithm and Fig.4, we have seven (0–6) priority levels in our example. We should treat prioritized packets differently while transmitted over the DiffServ network. The mechanism of priority mapping from priority to DiffServ PHB we adopt in our simulation is listed in Tab.4. Such a mechanism can ensure the transmission of all the base layers and part of the enhancement layers, in order to get an acceptable video quality. If possible, the best-effort packets can also be delivered to the receivers. Fig.5 depicts the general system architecture for prioritized AVOs transmission over the DiffServ.

<table>
<thead>
<tr>
<th>Priority-mapping to DiffServ PHB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1-3</td>
</tr>
<tr>
<td>4-6</td>
</tr>
<tr>
<td>DiffServ PHB</td>
</tr>
<tr>
<td>EF PHB</td>
</tr>
<tr>
<td>AF11</td>
</tr>
<tr>
<td>AF12</td>
</tr>
<tr>
<td>Best Effort</td>
</tr>
<tr>
<td>AF13</td>
</tr>
</tbody>
</table>

Fig.5 System architecture of the server and client side

5 Simulation Results

The MPEG-4 traffic is generated from the digital camera input as shown in Fig.3, and the simulations are conducted using $ns^2$\cite{15}. For convenience, there are only scaling and a little panning of the camera while capturing the scene. We manually extracted the AVOs from the scene and generated the video trace file for $ns^2$ simulation.

Fig.6 shows the network topology model used to generate the DiffServ network in $ns^2$ simulation. The
nodes R0-R3 are DiffServ-enabled router, i.e., the edge router and the core router. The video flow should compete with other TCP flows such as FTP, HTTP, etc. competing with other traffic types. On the contrary, while transmitted over the DiffServ with prioritization, the higher priority packets are more likely to be transmitted, i.e., more important information in the VOs are protected in order to obtain a better visual quality.

The presentation quality (in PSNR) of each frame received is also compared with the original video quality (average PSNR: 34.2 dB) and the ‘best-effort’ quality (average PSNR: 26.3 dB) in Fig.8. The visual quality of our scheme (average PSNR: 30.8 dB) outperforms most of the time the ‘best-effort’ scheme, even under stringent network condition, as shown in Fig.6, which also validates the effectiveness of the OPA algorithm.

6 Conclusions

We have given an efficient scheme for prioritized MPEG-4 audio-visual objects transmission over the DiffServ network, based on initial priority value assigned to each AVO by content provider. Simulation results show our method is effective. The further research in this direction will focus on the more precise control of the video streaming over the Internet, by considering multi-QoS parameters such as the packet’s delay and loss probability, etc.
References


Brief Introduction to Author(s)

HUANG Tian-yun (黄天云) was born in 1973. He received his M.S. degree in applied mathematics from Chongqing University, China in 1999, and Ph.D. degree in computer science from University of Electronic Science & Technology of China in 2003. His research interests include distributed multimedia system and computer networks, especially issues and technologies of multimedia flow modeling, analysis, transmission, and QoS control over Internet.

ZHENG Chan (郑 婵) was born in 1978. She received her M.S. degree in computer science from University of Electronic Science & Technology of China in 2003. Her current interests are software engineering and computing complexity.