MPEG-4 FGS Video Streaming Traffic Delivery Experimentation in an IP/DVB Network

Thomas Pliakas¹, George Kormentzas¹, Charalabos Skianis², Anastasios Kourtis²

¹ University of Aegean, Department of Information and Communication Systems Engineering, GR-83200, Karlovassi, GREECE
{tpliakas, gkorn}@aegean.gr

² National Centre for Scientific Research ‘Demokritos’, Institute of Informatics & Telecommunications, Aghia Paraskevi Attikis, POB 60228, Athens, GREECE
{skianis, kourtis}@iit.demokritos.gr

Abstract— The paper targets to demonstrate through a set of experimental studies that the common operation of IP DiffServ and DVB Bandwidth Management (BM) mechanisms can offer quality gains for prioritized MPEG-4 FGS media delivery across an heterogeneous IP/DVB setting. The experimental studies refer to the delivery of eight YUV QCIF 4:2:0 different video sequences across a heterogeneous IP/DVB testbed that includes two IP autonomous systems interconnected through a DVB MPEG-2 autonomous system acting as a trunk network.

Keywords—Bandwidth Management; DiffServ; MPEG-4 FGS; Packet Prioritization; Quality Metrics

I. INTRODUCTION

The paper discusses a heterogeneous cluster of networks comprising of two IP and one DVB domains. The developed testbed encompasses DiffServ technology in the IP domain and bandwidth management over MPEG-2 Transport Stream in the DVB domain. The IP domains are realized with PCs running Linux Operating System acting as border and core routers. These Linux-based routers are employing DiffServ capabilities implemented through open source software. The DVB domain is realized through commercial equipment patched with the ability to discriminate the DiffServ traffic aggregates and apply bandwidth management. The major goal of these experiments is to demonstrate that the common operation of IP DiffServ, DVB BM mechanisms and scalable MPEG-4 FGS prioritized video streaming offer quality gains for continuous media applications.

The paper is organized as follows: Section II discusses in detail a Linux-based heterogeneous IP/DVB testbed for MPEG-4 FGS video streaming traffic delivery experimentation. The testbed configuration details for the media delivery experimental studies and the results of these studies are discussed in Sections III and IV respectively. Finally, Section V draws conclusions and discusses directions for further work and improvements.

II. A LINUX BASED HETEROGENEOUS IP/DVB TESTBED FOR MPEG-4 FGS VIDEO STREAMING TRAFFIC DELIVERY EXPERIMENTATION

Figure 1 depicts the implemented heterogeneous IP/DVB testbed for MPEG-4 FGS media delivery experimentation studies. The testbed includes two IP autonomous systems interconnected via a DVB MPEG-2 autonomous system, acting as a trunk network.

The implementation of both IP and DVB autonomous systems is mainly based on open source software. The choice of open source software enables the configuration of the testbed according to the needs and the specific requirements of the experimental studies that are carried out. The major goal of these experiments is to demonstrate that the common operation of IP DiffServ, DVB BM mechanisms and scalable MPEG-4 FGS prioritized video streaming offer quality gains for continuous media applications. The next three subsections discuss implementation details of DiffServ, BM mechanisms and MPEG-4 FGS coding and packetization approaches, respectively.

A. IP Domain – DiffServ Implementation

The Differentiated Services (DiffServ) [1] framework aims to provide service differentiation within backbone IP networks. DiffServ technology enables the deployment of IP traffic discrimination in a scalable manner, by providing QoS...
guarantees only for aggregated traffic classes rather than for specific flows. Essentially, when entering a network, packets are placed into a broad service group by a classification mechanism that reads the DiffServ Code Point (DSCP) in the IP packet header and the source and destination address. The advantage of such a mechanism is that several different traffic streams can be aggregated to a small number of Behavior Aggregates (BA), thereby simplifying processing and associated storing and forwarding processes at the routers. Furthermore, there is no need for signaling and the traffic differentiation is obtained on a packet-by-packet basis.

Differentiated Services model as proposed by IETF support two different services: (1) the Expedited Forwarding (EF) that supports low loss and delay/jitter, and (2) the Assured Forwarding (AF) that provides better QoS than the best effort, but without guarantee. For streaming video applications, in which the encoding and decoding process is more resilient to packet loss and delay variations, besides Premium Service, the Assured Service can be employed. Note that MPEG-4 FGS originally assumes guaranteed delivery to Base Layer (BL) and leaves the Enhancement Layer (EL) to the mercy of best effort service of Internet.

The international literature presents a number of DiffServ implementations [2][3]. However, most of them are poorly documented and/or quite outdated. In this context, we implemented our own version running over Linux OS with kernel version 2.6.11 [4]. Our implementation follows the generic architecture of Figure 2. Specifically, according to a set of filters, the incoming packets are separated into a number of classes, where each one of them maintains its own queuing/scheduling discipline for serving its packets, as well as its own policing scheme for controlling the amount of its packets.

![Figure 2. Linux DiffServ Architecture](image)

The implemented DiffServ mechanism is incorporated into the two IP autonomous systems of the heterogeneous IP/DVB testbed. Each autonomous system consists of three PCs (at least PIII CPU with 512MBytes of RAM) running Linux OS (kernel version 2.6.11) with iproute2 package and tc utility support. Each IP domain includes two edge routers and one core router. The supported BAs are EF, AF1x and BE. The Hierarchical Token Bucket (HTB) packet scheduler with three leaf classes is used for the realization of the supported BAs. In particular, a pFIFO queuing discipline is adopted for the EF BA. Three GRED virtual queues with different drop percentages are implemented for the AF1x BA. The BE BA is served through a RED queuing discipline. This setting is depicted in Figure 3. The maximum bandwidth allocated at the parent HTB class is 13Mbps shared among the BAs. Each leaf class can borrow excess bandwidth from another leaf class.

The configuration of the GRED virtual queues requires the adjustment of the following parameters: $AvgQ_{\text{max}}$ which is the maximum average queue size after which all packets get dropped, BS, which is the percentage of the bandwidth share, $L$, which is the desired latency, BW, which is the total link bandwidth, $AvgQ_{\text{max}}$ which is the minimum average queue length after which packets get dropped, $AvgPkt$, which is the average packet size, $B$, which is the burst value in number of packets and $Q_{\text{limit}}$ which is the actual queue length never to be exceeded.

![Figure 3. Linux DiffServ implementation](image)

The Average Packet (AvgPkt) size is 1024 bytes. The percentage of the Bandwidth Share (BS) is 33%. The desired latency $L$ is either equal to 100ms for all AF1x BAs, or 100msec for AF11, 200msec for AF12 and 500msec for AF13 BA. The remaining parameters are calculated based on the following simple formulas (1) – (4) and the corresponding results are given in Table I:

$$AvgQ_{\text{max}} = \frac{0.01 \cdot BS \cdot L \cdot BW}{8 \cdot \text{bits} \cdot 1000 \cdot \text{ms} \cdot \text{bytes} \cdot \text{sec}}$$  \hspace{1cm} (1)

$$AvgQ_{\text{min}} = 0.5 \cdot AvgQ_{\text{max}}$$  \hspace{1cm} (2)

$$B = 2 \cdot AvgQ_{\text{max}}$$  \hspace{1cm} (3)

$$Q_{\text{limit}} = 4 \cdot AvgQ_{\text{max}}$$  \hspace{1cm} (4)

<table>
<thead>
<tr>
<th>Set</th>
<th>BA</th>
<th>L</th>
<th>AvgQ_{\text{max}}</th>
<th>AvgQ_{\text{min}}</th>
<th>B</th>
<th>Q_{\text{limit}}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>AF11</td>
<td>100</td>
<td>162500</td>
<td>54167</td>
<td>90</td>
<td>433336</td>
</tr>
<tr>
<td></td>
<td>AF12</td>
<td>100</td>
<td>162500</td>
<td>54167</td>
<td>90</td>
<td>433336</td>
</tr>
<tr>
<td></td>
<td>AF13</td>
<td>100</td>
<td>162500</td>
<td>54167</td>
<td>90</td>
<td>433336</td>
</tr>
</tbody>
</table>

B. DVB Domain – BM Implementation

The bandwidth reallocation among the IP virtual channels of a DVB MPEG-2 TS [5][6] uplink is based on a set of predefined priority policies. In this work three priority policies...
are implemented, namely: (1) **Static guaranteed** - This policy guarantees a static bandwidth to each virtual channel. A guaranteed bit rate value has to be specified so that the actual bit rate is guaranteed up to this boundary value. The unused bandwidth (guaranteed bit rate - instant bit rate) is reserved and cannot be allocated to other virtual channels. (2) **Dynamic guaranteed** - This policy guarantees a dynamic bandwidth to each virtual channel. A guaranteed bit rate value has to be specified so that the actual bit rate is guaranteed up to this boundary value. On the contrary to the static guaranteed policy, the unused bandwidth (guaranteed bit rate - instant bit rate) is not lost, but can be allocated to other virtual channels. (3) **Best effort** - This conventional policy allocates bit rate to various virtual channels based on the available bandwidth.

Two full uplink/downlink configurations comprise the DVB domain. The uplink involves an encapsulator, a multiplexer and a DVB modulator. The downlink is realized via a DVB/IP gateway, which is a standard PC running Linux operating system equipped with a standard Ethernet controller and a DVB PCI card capable of demodulating the DVB signal and de-encapsulating the IP packets. The DVB domain employs the implemented priority policies in order to preserve BAs defined in the IP domains. The binding among BAs and the corresponding priority policies is given in Table II:

<table>
<thead>
<tr>
<th>BA</th>
<th>Priority policy</th>
<th>Guaranteed bit rate</th>
<th>Maximum bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>EF</td>
<td>Static</td>
<td>3.6Mbps</td>
<td></td>
</tr>
<tr>
<td>AF11</td>
<td>Dynamic</td>
<td>2Mbps</td>
<td>14Mbps</td>
</tr>
<tr>
<td>AF12</td>
<td>Dynamic</td>
<td>2Mbps</td>
<td>14Mbps</td>
</tr>
<tr>
<td>AF13</td>
<td>Dynamic</td>
<td>2Mbps</td>
<td>14Mbps</td>
</tr>
<tr>
<td>BE</td>
<td>Best Effort</td>
<td>2Mbps</td>
<td>14Mbps</td>
</tr>
</tbody>
</table>

Note that in order to deal with the IP to MPEG-2 encapsulation overheads, the total link bandwidth is 14 Mbps, which is 1 Mbps bigger than the IP domains’ one. While AF1x BAs and BE BA can borrow bandwidth beyond the guaranteed, the EF BA is statically allocated a maximum value and therefore cannot borrow unused bandwidth.

### C. MPEG-4 Video Encoding and Packetization

MPEG-4 FGS [7] scalable video coding constitutes a new video coding technology that increases the flexibility of video streaming. Similar to the conventional scalable encoding, the video is encoded into a Base Layer (BL) and one or more (ELs). For MPEG-4 FGS, the EL can be efficient truncated in order to adapt transmission rate according to underlying network conditions. This feature can be used by the video servers to adapt the streamed video to the available bandwidth in real-time (without requiring any computationally demanding re-encoding). In addition, the fine granularity property can be exploited by the intermediate network nodes (including base stations, in case of wireless networks) in order to adapt the video stream to the currently available downstream bandwidth. In contrast to conventional scalable methods, the complete reception of the EL for successful decoding is not required [8]. The received part can be decoded, increasing the overall video quality according to the rate-distortion curve of the EL as described in [9][10].

The most widely used scheme, in order to packetize MPEG-4 video streams, is fixed-length packetization, where video packets of similar length are formed. The packet size of video stream is also related to efficiency and error resiliency because a smaller packet size for example requires a higher overhead but has a better performance in error prone networks. By evaluating the expected loss impact of each packet to the end-to-end video quality, we can assign priority to each packet according to its importance in video sequence. With assigned priorities, the packets are sent to underlying network and receive different forwarding treatments.

### III. TESTBED CONFIGURATION

Eight YUV QCIF 4:2:0 color video sequences consisting of 300 to 2000 frames and coded at 25 frames per second are used as video sources. Each group of pictures is structured as IBBPBBPBB… and contains 25 frames, with maximum UDP packet size of 1000 bytes (payload only). The Microsoft MPEG-4 FGS encoder/decoder is used for encoding YUV sequences. A number of background flows is transmitted in the network, in order to lead the DiffServ/DVB system in congestion. The background traffic is always running and is assigned to the BE BA. The latter BA has always the following characteristics: Poisson distribution with 1472 bytes packet size and constant rate of 8 Mbps. Correspondingly, EF BA is generated at 3Mbps rate and each AF1x BA at 2Mbps rate. The assigned bandwidth to AF1x BA is equally segmented to support three-drop percentages, which are 2% for AF11, 4% for AF12 and 6% for AF13.

The transmitter and the receiver reside on the same system (PC) in order to avoid issues that arise from synchronization errors or/and differences in system clocks [11]. The video traffic is transmitted from the source network interface, which is connected at the ingress router of autonomous system AS1, passes through the three different network domains and is finally returned back to the source system. For each generated packet, identified by a unique sequence number, the departure and arrival timestamps, and the type of payload that contains, are obtained. When a packet does not reach the destination, it is counted as a lost one. It is not only of interest the amount of lost packets, but also the type of content that packets have in their payload. Furthermore, not only the actual loss is important for the perceived video quality, but also the delay of packets/frames and the variation of the delay, usually referred to as packet/frame jitter. The packet/frame jitter can be addressed by so called play-out buffers. These buffers have the purpose of absorbing the jitter introduced by the network delivery delays. It is obvious that a big enough play-out buffer can compensate any amount of jitter. There are many proposed techniques in order to develop efficient and optimized play-out buffer, dealing with this particular trade-off. These techniques are not within the scope of the described testbed. For our experiments the play-out buffer is set to 1000 ms.

In order to measure the improvements in video quality by employing MPEG4 FGS prioritized video streaming, we adopt the **Peak Signal-to-Noise Ratio (PSNR)** and **Structural
\[ PSNR(n_1,n_2) = 10 \log \frac{V^2}{MSE(n_1,n_2)} \] (5)

where \( V \) is the maximum value of a pixel (255 for 8-bit grayscale images) and \( MSE(n_1,n_2) \) is defined as:

\[ MSE(n_1,n_2) = \frac{1}{XY(n_2 - n_1 + 1)} \sum_{x=1}^{n_2} \sum_{y=1}^{n_1} (I(x,y,n) - \bar{I}(x,y,n))^2 \] (6)

where \( M \) is given by:

\[ M = \left[ I(x,y,n) - \bar{I}(x,y,n) \right] \] (7)

and \( I \) is the luminance component of source image and \( \bar{I} \) is the luminance component of destination image. Note that, both PSNR and MSE are well defined only for luminance values. As mentioned in [12], the human visual system is much more sensitive to the sharpness of the luminance component than that of the chrominance component [15], therefore, we consider only the luminance PSNR (Y-PSNR).

SSIM is a novel full reference metric for measuring the structural similarity between two image sequences, exploiting the general principle that the main function of the human visual system is the extraction of structural information from the viewing field. If \( x \) and \( y \) are two video signals, then SSIM is defined as:

\[ SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\mu_x^2 + \mu_y^2 + \sigma_x^2 + \sigma_y^2 + C_1} \] (8)

where \( \mu_x, \mu_y, \sigma_x, \sigma_y \) are the mean of \( x \), the mean of \( y \), the variance of \( x \), the variance of \( y \), and the covariance of \( x \) and \( y \), respectively. The constants \( C_1 \) and \( C_2 \) are defined as:

\[ C_1 = (K_1L)^2 \] (9)

\[ C_2 = (K_2L)^2 \] (10)

where \( L \) is the dynamic range of pixel values and \( K_1 = 0.01 \) and \( K_2 = 0.03 \) respectively.

IV. EXPERIMENTAL RESULTS

This section evaluates the performance of the proposed testbed configuration through a set of four experimental cases. We study the performance of our framework by enabling or disabling scalable video coding, or by enabling or disabling prioritized transmission. The quality gains of scalable video coding in comparison with non-scalable video coding and the quality gains of prioritized transmission in comparison with non-prioritized transmission are compared in detail.

The first experiment refers to a single layer MPEG-4 video stream transmission, where both DiffServ and BM mechanisms are not applied to the heterogeneous IP/DVB testbed.

The second experiment refers to a scalable MPEG-4 FGS stream transmission of two layers, with both DiffServ and BM mechanisms deployed to the heterogeneous IP/DVB testbed. The BL packets are encoded using the MPEG4-FGS codec with MPEG2-TM5 rate control at 256 Kbps and the EL one encoded at 256 Kbps. By assigning high priority, Premium Service (EF) to BL, we can guarantee proper reception of the BL and without losses. For the EL, we assign priorities according to the anticipated loss impact of each packet on the end-to-end video quality (considering the loss impact to itself and to dependencies). Each layer has a priority range, and each packet has different priority according to its payload. The packets, which contain data of an I-frame are marked with the lowest drop probability (AF11), the packets which contain data of a P-frame are marked with medium drop probability (AF12) and the packets which contain data of a B-frame are marked with high drop probability (AF13).

The third experiment refers to a scalable MPEG4 video stream transmission consisting of one BL and two ELs (i.e., EL1 and EL2). The encoding of BL packets remains at 256 Kbps as in the second case, while the encoding of packets of both ELs is at 256Kbps. For this case, we use EF for transmitting BL, AF11 for transmitting EL1, and Best Effort (BE) for transmitting EL2. For this case both DiffServ and BM mechanisms are active as in the second experiment.

Finally, the fourth experiment adopts the setup of the third case, while it applies the prioritized packetization scheme of the second experiment to the packets of the first EL (i.e., for this case, we use EF for transmitting BL).

Table III depicts the experimentation results in terms of PSNR and SSIM video quality metrics for eight different video sequences. It is obvious that for the second experimentation there is a significant gain in video quality of 2.3dB in terms of PSNR when compared to the first scenario. In some video sequences with many differences between scenes the video quality gain is more that 3dB. Furthermore, in the third experiment we can observe a gain in video quality of 1.2dB, compared to the second experiment. At the fourth scenario, the video quality, in terms of PSNR, remains at the same level.
For the *Highway* video sequence (consisting of 2000 frames), we measure the packet/frame losses for I-, P-, and B-frames for the four experimental cases with results presented in Table IV.

**TABLE IV. DETAILED RESULTS FOR THE HIGHWAY VIDEO SEQUENCE**

<table>
<thead>
<tr>
<th>Frame Type</th>
<th>Delay (ms)</th>
<th>Jitter (ms)</th>
<th>Loss (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Packet</td>
<td>Frame</td>
<td>Packet</td>
</tr>
<tr>
<td>I</td>
<td>954.21</td>
<td>981.43</td>
<td>9.37</td>
</tr>
<tr>
<td>P</td>
<td>923.43</td>
<td>972.32</td>
<td>9.18</td>
</tr>
<tr>
<td>B</td>
<td>973.82</td>
<td>961.32</td>
<td>9.32</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>302.89</td>
<td>323.21</td>
</tr>
<tr>
<td>P</td>
<td>340.82</td>
<td>323.67</td>
<td>7.32</td>
</tr>
<tr>
<td>B</td>
<td>942.31</td>
<td>969.23</td>
<td>7.58</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>299.96</td>
<td>319.21</td>
</tr>
<tr>
<td>P</td>
<td>304.94</td>
<td>325.74</td>
<td>6.82</td>
</tr>
<tr>
<td>B</td>
<td>301.67</td>
<td>323.43</td>
<td>6.86</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>303.61</td>
<td>312.21</td>
</tr>
<tr>
<td>P</td>
<td>338.43</td>
<td>347.72</td>
<td>7.74</td>
</tr>
<tr>
<td>B</td>
<td>923.44</td>
<td>969.23</td>
<td>7.89</td>
</tr>
</tbody>
</table>

By isolating the losses and the delays to P- and B-frames we can achieve significant gains to video quality. Packet losses, which P-frame content, can affect not only the decoding process of P-frames but also the B-frames. This lead to higher percentages of B-frame losses but it is a significant affect to the overall video quality. In the fourth scenario, the user can achieve the same video quality, compared to third scenario, without using only the AF11 traffic class of the DiffServ. By distributing the traffic to all traffic classes, achieving the same video quality, in the lowest price, by sending lowest traffic to the cost effective AF11 traffic. From the network provider perspective, the provider’s network can use more efficient its bandwidth, by serving more users, at the level of quality they pay.

**V. CONCLUSIONS**

The paper shows that the common operation of IP DiffServ and DVB BM mechanisms can offer quality gains for media delivery across heterogeneous IP/DVB settings. In this context, this study could constitute a potential vehicle for end-to-end QoS provision. Towards this purpose, the paper presents experimental results of an empirical study of a Linux-based heterogeneous IP/DVB network supporting continuous media applications. The development of new service categories increases the need for a differentiated, at the network level, treatment of the information packets, based on their different association with each type of service. This brings forward the concept of differentiated QoS provisioning, that is, the possibility to guarantee the most suitable service level for every traffic category.

Several issues remain open and are currently under research. For example, a more efficient mechanism for prioritized packetization of video bit stream is required. Moreover, the distribution of packet priority and a price mechanism according to DS level remains to be examined.

**ACKNOWLEDGMENT**

The work reported in this paper is carried out within the project ”Study and Development of Interactive Broadband Services based on DVB-T/DVB-H Technologies” in the context of framework 2.2 of ‘Pythagoras II – Research Group Support of the University of the Aegean’ jointly funded by the European Union and the Hellenic Ministry of Education.

**REFERENCES**


