Content-Aware Adaptive Video Streaming System

Magdy A. Koutb, Hamdy M. Kelash, Reda H. Aboelez, and Mohammad A. Talaat

M. A. Koutb
Faculty of Electronic and Electrical Engineering, University of Menofeya, Menouf, Egypt.

H. M. Kelash
The Computer Sciences and Engineering Department,
Faculty of Electronic and Electrical Engineering, University of Menofeya, Menouf, Egypt.

R. H. Aboelez
The Computer Engineering Department, University of Alazhar, Cairo, Egypt.

M. A. Talaat
The Computer Sciences and Engineering Department,
Faculty of Electronic and Electrical Engineering, University of Menofeya, Menouf, Egypt.
e-mail: moadly@iti-idsc.gov.eg.

Abstract: Since there are no Quality of Service QoS guarantees for video streaming over best-effort IP networks, adaptation for both the audio and video streams of an established real-time streaming session must be applied to respond to network congestion conditions. In many video streaming applications, either the audio or the video stream for the same audio/video streamed content is more semantically important than the other. Hence it is better to let this stream, in cases of congestion, suffer less degradation in quality even if that imposed the other stream to suffer more degradation. This paper proposes a simple but efficient application-level content aware adaptive video streaming system that is primarily configured by the previously noted more semantically important stream. The system monitors the end-to-end network congestion level. In congestion cases the system degrades in steps the quality of the less important stream first. Then it moves, if necessary, to the other stream to degrade, according to a predefined adaptation mechanism. The system then triggers when the congestion case is over in order to start upgrading the degraded streams gradually back to their initially established states if the network conditions permit. This new concept in adaptation, when tested, lead video streaming applications users to be more satisfied with Internet video streaming services.

Keywords: Best-Effort, QoS, RTP, Video Streaming

1. INTRODUCTION

With the rapid advances in computer and network technologies, especially with the emergence of Internet, audio/video streaming applications are becoming very popular. Most importantly the low cost of using computer besides the recent progress of broadband access...
lines, such as xDSL and FTTH, make video streaming applications have a great number of potential users. However, video streaming technology is still far from perfection due to the technical problems this technology faces, and the drawbacks in their currently proposed solutions. The heterogeneous network environment that Internet provides to the real time applications as well as the lack of sufficient QoS guarantees in best-effort networks, many times enforces applications to embody adaptation in order to enhance their performance. Otherwise the transmission will encounter high packet loss rates during periods of insufficient network bandwidth. Furthermore the high packet loss will result in continuous content quality degradation due to error propagation. Adaptation is achieved by adjusting the quality of the streamed audio/video to match the underlying available network bandwidth.

QoS is considered a good solution. It is generally divided into two main categories; Network-level which is mainly based on router configuration to assure a certain packet loss, delay, and jitter, and Application-level which builds specific applications that make end to end adjustments. The big problem that makes QoS not always the perfect solution and allows adaptation to appear as a more feasible one lies in the large number of different administration domains composing Internet; the QoS strategy implemented in a domain that may be different than that in the neighboring one. For example a domain may be implementing QoS based on DiffServ Concept [1], another using Intserv Concept [2], a third built on IPv6 infrastructure [3], and a fourth not applying QoS at all. A given stream can not be able to pass these four domains in any sequence keeping its QoS constraints due to lack of compatibility between their QoS rules. Also QoS lacks the feature of being TCP friendly while TCP forms the majority of the coexisting Internet connections in most of channels. A definition for the TCP friendly connection is in [4]. The multicast mechanism of the internet allows a single video streaming source to direct and distribute any data stream to multiple receivers without the need to replicate data on single connections to each receiver, and consequently without increasing the network bandwidth, as in the traditional unicast (one-to one) case. The heterogeneity problem arises when a single video streaming source transmits a session to many receivers which are diversified in their network connections capabilities whether good or poor. Even when QoS was applied JQoS [5] tackled the heterogeneity problem by introducing an adaptation mechanism in order to help in tuning the source streamed bit rate so that high bandwidth connection receivers get a better quality version of the session being streamed meanwhile the low bandwidth connection receivers get the best quality possible version that their connection tolerates of the same session concurrently.

On the other hand, adaptation applied in many systems, like in [6]–[8], missed an important issue in our point of view. This point is the Content-Awareness property that must exist in such systems, which means that the adaptation system should be aware of the application it is embedded in, and consequently the content being streamed in order to make right quality degradation decisions on the right media when needed. For example in distant learning applications, the content being streamed can be educational lectures given by a lecturer in a student hall room. The video stream of this content will convey only the lecturer visage, mouth motion, and facial gestures, while the audio stream content is the real semantically relevant stream for the students receiving this session. The audio stream of such audio/video session conveys the scientific material intended to be delivered to students. The student
receiving this session remotely would logically prefer if all the necessary quality degradation performed by the adaptation mechanism was applied only on the video stream while keeping the audio stream quality untouched as far as possible. Another example would be if this lecturer decided to show his students a surgical operation he is making, while the students are watching it remotely through video streaming receiving applications, in this case the video content of the audio/video session would convey the more semantically relevant data intended to reach the students, while the audio stream is not of the same importance. Hence the student in this case also would prefer that any quality degradation necessary be directed to the audio stream; meanwhile the video stream maintains its best available quality for the longest possible interval of time.

The system proposed in this paper is said to be Content-Aware when applying its application-level adaptation mechanism in response to congestion. It is primarily configured by the more semantically relevant stream option in the session to be streamed, prior to the session start, in order to build its adaptation mechanism steps based on this option. It implements quality control for audio/video streaming systems over IP best-effort networks by taking the advantage of continuous media compression techniques, the real-time transport protocol RTP/RTCP[9], the Sun Microsystems' Java Media Framework (JMF) [10], and Remote Method Invocation (RMI) technology. The rest of this paper is organized as follows: The system architecture is shown in section II, its module design implementation is discussed in section III, and section IV demonstrates the testing performed and its results. Finally conclusions and future work are presented in section V.

2. SYSTEM ARCHITECTURE

The main design goal of this system lies firstly in adapting the bit rate of both the audio and video content streams of a given streaming source in response to congestion occurrences, secondly in making this quality degradation according to a specified adaptation mechanism that both streams will follow in a step manner given that one of the streams is primarily defined to be the more semantically relevant, and namely it is the last to be degraded if needed.

A typical Internet video streaming system is shown in Fig. 1 and it works as follows: the
system is composed of a source and a receiver, or a group of receivers, which all become members of a multicast group by joining this group using a multicast IP address and a given port number. The source sends to the receiver its compressed live audio and video content for the streamed session each on a separate RTP stream. The receiver is capable then of receiving the RTP streams and playing them back. During the session time each receiver issues a series of RTCP reports for each received stream periodically. These reports are destined to the same multicast IP address and port number. They help in identifying the most recent receiving status of this receiver mainly regarding jitter, the number of packets lost, and its fraction from the sent packets. The Session Manager presence is essential to avoid source overloading or crashing if the monitoring process done was also left to the source to handle. This Session manager logically exists between the source and the receiver and also joining the same multicast group. In order to trace and monitor the receiving status of receivers we benefited from the work in [11] to produce our RTP Monitor module. This module is responsible of analyzing the arriving RTCP reports issued by receivers, and focuses on the packets fraction lost parameter which is a good measure of congestion in the network path between the Source and a certain receiver as discussed in [5]. The RTP Monitor can report to the adaptation mechanism a congestion alert which is a Boolean value that signifies a congestion case presence or its absence when passed. The Relevant Stream Option is an externally introduced input by the system user. It must be supplied before the start of every session. This option simply informs the system whether the audio or the video stream is the more semantically relevant stream for this session. Accordingly it may stay unchanged for a group of like streamed contents in a number of consecutive sessions. The Controller produces the proper adaptive decision by either improving or degrading the quality of a certain RTP stream based on both the Congestion Alert value and the Relevant Stream Option and according to a predefined Adaptation Mechanism.

The challenge in system design is to establish a proper dynamic quality Adaptation Mechanism. To demonstrate the problem we will thoroughly explain our Adaptation Mechanism in the next paragraph.

2.1. Adaptation Mechanism

The whole system is very much analogous to a feedback control system. Firstly we must notice that each compression technique for either the audio or the video stream can not be granularly increased or decreased in its produced bit rate, and consequently in its quality level. Thus it can enable either quality improving or degrading in defined gap steps. In our system we typically found that three degraded versions of each stream can be available in most of cases, since most of the encoding techniques JMF supports can handle only three versions and not much more. These versions produce by decreasing the stream bit rate by known gap values. For example in the DVI Audio Encoding algorithm, JMF support the sample rate values: 8 KHz, 11.025 KHz, or 22.050 KHz using 16 bits/sample [10].

To demonstrate, we considered a given example content that is needed to be streamed. The video RTP stream in this content is the more relevant. Fig. 2 shows the Adaptation Mechanism applied on the previous content to produce the proper adaptive decisions sent to the streaming source, which in turn produces its adapted stream to network. The RTP Monitor
of the system is shown to be receiving the RTCP reports from receivers, when these reports show congestion in the network path between the source and a specific receiver by showing fraction packets lost more than 0.05 as a typical value, a Boolean variable named as the congestion alert is set to true and sent to the system Controller. The Controller works based on two inputs this alert is one of them, and the other is the external option notifying it by the more relevant stream for this specific content. Two integer variables are defined which namely are \( D_A \) and \( D_V \). These variables represent which degraded version of the audio or the video streams are currently being transmitted over the network consequently. Their values for lie between 0 and 3 for both since both streams have three degraded versions as previously mentioned. Zero value for any of them means that the corresponding stream is currently being transmitted by the source without any degradation applied on it. Each time a more degraded version is decided to be transmitted for any of the streams its \( D \) value is incremented by one. On the other hand decrementing \( D \) by one means that the more better quality version of the stream directly above the current will be sent. You can see in Fig. 2 that the video stream of this session will suffer no degradation due to congestion alerts unless it is made sure that the audio stream is currently in its least degraded version. This is of course due to the relevant stream option supplied to the controller that prioritized the video stream of this content.
Fig. 2. Adaptation Mechanism

It should be hinted that in case of multiple receivers' presence in the system, the controller may suffer quality oscillations. This problem was previously handled in [12] which proposed a smoothing equation that can be referred to in order to avoid this problem.

3. MODULE DESIGN

Our system is demonstrated using the Java Media Framework (JMF) as a media-developing tool. JMF has the advantage of implementing RTP/RTCP functionalities in real time using friendly APIs and saving the effort of building these capabilities from scratch. We can take a closer look to the implementation of each of our system modules which are the Streaming Source, Receiver, and Session Manager composed of RTP Monitor and Controller in the coming paragraphs.

3.1. Streaming Source

Fig. 3 shows the RTP stream producing steps in JMF.

The required components to stream an audio/video session are namely a Data Source, Processor, and RTPSessionManager. The Data Source can either be a real-time captured media or stored media file, the processor is used to encode this Data Source and then the role of the RTPSessionManager is to manage the established session. Our mission was to make our streaming source real time adaptive by adding to it processing transmission controls. These controls enable changing the media encoding parameters without the need to re-establish the session. The code used in this job contained the JMF TrackControl class and its getTrackControls() and getFormat() methods. Hence after obtaining the transmission controls, one can build the adaptation mechanism equipped with the necessary RMI interfaces that enables the remote object, which is the Controller of the Session Manager in our system, to call. Taking example encoding algorithms for both audio and video used in our system we would mention G273 and GSM for audio streams and H263 for video streams. JMF supplied functions for audio streams such as: BitRateControl() and SilenceSuppressionControl() and for video streams it supplies functions such as: BitRateControl(), KeyFrameControl(), QualityControl, and PacketSizeControl().

3.2. Receiver

Fig. 4 shows both the audio and video players of JMF. JMF provides a separate player for each of the RTP audio and video streams. The video player is on the right and the audio is on the left section of the figure.
Fig. 4. Audio and Video Players

The player is created through using the `createPlayer()` method provided in the `SessionManager`. This method supplies a processor for the received streams to enable playing. It is worth mentioning that JMF version 2.1 players contain built in quality monitoring facilities. Players can popup another window that displays the transmission statistics report of the stream being played. You will be able to read in this report values such as: Received Packets, Received Bytes, Lost PDUs…etc.

### 3.3. Session Manager

The Session Manager module of our system is divided into the RTP Monitor and the Controller sub-modules.

1) **RTP Monitor:** A typical RTP Monitor function is to collect the receivers *Receiver Reports* sent by multicast session receivers abbreviated as *RR* and show their information in a readable form. Receiver Report contains information about the three main parameters representing a stream quality which are: jitter, delay, and fraction of packet lost. The RTP Monitor Embodied in our system managed to show a block report for all receivers of a given multicast session showing the mentioned values as in Fig. 5.

![Fig. 5. Receivers’ Feedback Reports in the RTP Monitor](image)

Receiver reports contain fields about other information that can be just merely mentioned such as: Local Collisions, Remote Collisions, and Looped Packets. JMF implements an interface named `RTCPReport` that can be a `SenderReport` or `ReceiverReport` also there is the `getFeedback()` method that returns a vector of `RTCPFeedback` objects. Each of these objects corresponds to one of the session receiver's information. We added to the monitor the capability to analyze the fraction packets lost figure found in the Receiver Report and comparing it to a certain threshold in order to set the Boolean congestion alert variable. We took our threshold value as 0.05 fraction lost packets which if passed signifies congestion presence at the receiver side.

2) **Controller:** The system controller acts on the basis of the congestion alert variable value, and the externally supplied option which either notifies the controller that the audio stream is more semantically relevant for this session or the video stream. This option is supplied to the...
controller through two simple interface radio buttons labeled as Audio and Video. The controller flow-chart can be seen in Fig. 2 and no need to repeat it here. The controller produces an adaptive decision that is always one of four: degrade audio, degrade video, upgrade audio, or upgrade video. The decision is always one step degrade or upgrade. The controller may also be in hold state in two, which means that there is no proper adaptive decision produced in these two cases. This first case is when both versions are in their top upgraded versions and meanwhile no congestion alert with true value is reported, and consequently no degradation for either is needed. The second case is when both streams reach their third, and last, degraded version due to the arrival of group of true consequent congestion alerts, here the controller can not send an additional degrading decision since there is no degraded version available for any of the streams to switch to. Both streams, in this case, are left for the network default adaptation mechanism which is built on discarding the lastly arrived set of packets which the network nodes queuing capacity can not afford. The system retains its control just at the time of arrival of the first false congestion alert. At this moment the controller starts to perform upgrading again and according to the adaptation mechanism as well. The code used in the controller implementation is simple. RMI utilized technology was used to achieve the connection between controller and source to invoke the sources methods for adaptation.

4. SYSTEM PERFORMANCE EVALUATION

In Fig. 6 we demonstrate our used test-bed environment. It is composed of five Personal Computers (PCs) connected through an isolated fast Ethernet hub and running Microsoft Windows XP Operating System.

![Test-bed Environment Diagram](image)

The Source PC uses either stored or real-time captured audio/video media to stream to network. This media normally passed by two live compression processes, one for audio and another for video, by one of the compression techniques supported by JMF. Two Receiver PCs joining, concurrently with the source the same multicast group, play back the streamed media through JMF players. The Session Manager PC features logically between the sender and the receiver connected to both of them. To simulate the congestion occurrence cases, the traffic generator PC to spark it when needed. The whole setup is believed to resemble the real-world Internet environment.

First we investigated the feasibility of the various audio video encoded formats that JMF
supports to be clear under reduced streaming bit rates. For the video formats we found that H.263/RTP video encoding algorithm is very tolerant to work under low bit rates, it managed to be clear in both still and motion pictures for the bit rates: 76 Kbps, 36 Kbps, and 16 Kbps. For the Audio formats the DVI/RTP was clear for 32 Kbps, G723/RTP for 6.325 Kbps and GSM/RTP for 13.44 Kbps.

The application of our content-aware adaptation concept is neither meant nor expected to present a type of an enhancement, over the previously implemented conventional adaptation systems, which can be measured and expressed in less delay time intervals for example. At the same time it is important for us also to show that it worked as good as they do regarding such issues. On the other hand our system was meant to highly achieve a far better user satisfaction with the video streaming service over Internet. Our system newly introduced content-awareness property performance was evaluated through a questionnaire process whose results are shown in Table 1.

Table 1 Questionnaire Results

<table>
<thead>
<tr>
<th>Media Type (Prioritized Stream)</th>
<th>Conventional Satisfaction</th>
<th>Content-Aware Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hall Room Lectures (Audio)</td>
<td>30 60 10 0 0 0 0 10 60 30</td>
<td></td>
</tr>
<tr>
<td>Surgical Operations (Video)</td>
<td>30 50 10 0 0 0 0 10 70 20</td>
<td></td>
</tr>
<tr>
<td>Soccer Goals (Video)</td>
<td>20 70 10 0 0 0 0 10 80 10</td>
<td></td>
</tr>
<tr>
<td>Movie Trailers (Video)</td>
<td>30 30 40 0 0 0 0 60 20 20</td>
<td></td>
</tr>
<tr>
<td>Interviews (Audio)</td>
<td>10 70 20 0 0 0 0 0 80 20</td>
<td></td>
</tr>
</tbody>
</table>

This questionnaire was designed to let a statistical sample composed of thirty of our colleagues and students, of Internet video streaming systems users express their degree of satisfaction with both conventional adaptation based systems which have no content awareness and our content aware system. This group of users played back a set of various media types which are usually streamed over LANs and WANs forming the Internet. After watching these media, once adapted by the conventional way, and another time adapted in our way, they expressed their degree of satisfaction with each media by an integer that ranges from one, which corresponds to poor satisfaction, to five which corresponds to excellent satisfaction. The number in each cell of the table represents the percentage of users who marked this cell in the questionnaire. Each media type relevant stream chosen is shown between brackets with the media type. Our choice for this option was built just on the logical semantic content of each type that imposes one of the audio or video to be the more relevant. The table shows that the content-aware system did achieve more user satisfaction for the chosen media types.

The technical evaluation of our proposed system showed no remarkable discrepancy than systems working without introducing the content-awareness feature. This is normal and was expected because each of the audio RTP stream and video RTP stream has a completely different encoding compression algorithm, so in both system types the system have to deal with each media stream, whether audio or video, separately to achieve adaptation.
shows a graph for a conventional system response of [5] and Fig. 8 shows our system response under the approximately the same testing conditions. The overall taken time by both systems to decay the packet loss ratio curve was almost the same. Hence we can say that there is no big difference between both responses regarding the time interval taken to adapt to congestion or low network resources. Hence we can say that the content-awareness had no technical negative effect.

Fig.7. Conventional System Response

Fig.8. Content-Aware System Response

5. CONCLUSION AND FUTURE WORK

This paper introduced a new content-awareness concept for video traffic adaptation in response to congestion over best-effort IP networks. The content aware adaptive system was demonstrated to be feasible and more user satisfactory. The new concept introduced had no effect on the system technical performance.

The media-developing tool used, JMF, is a user friendly and efficient one but has a drawback regarding the synchronization between audio and video streams. The automatic detection of the more relevant content stream can be considered as a future extension for this research; also further studies for its performance under multiple receivers' presence would be interesting.

REFERENCES
