INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2003; **16**:225–248 (DOI: 10.1002/dac.585)

Multimedia transmission with adaptive QoS based on real-time protocols

Ch. Bouras[1,2,*,†] and A. Gkamas[1,2,‡]

[1] *Computer Engineering and Informatics Department, University of Patras, GR-26500 Patras, Greece*
[2] *Research Academic Computer Technology Institute, 61 Riga Feraiou Str., GR-26221 Patras, Greece*

SUMMARY

In this paper, we describe a mechanism for adaptive transmission of multimedia data, which is based on real-time protocols. The proposed mechanism can be used for unicast or multicast transmission of multimedia data over heterogeneous networks, like the Internet, and has the capability to adapt the transmission of the multimedia data to network changes. In addition, the implemented mechanism uses an inter-receiver fairness function in order to treat the group of clients with fairness during the multicast transmission in a heterogeneous environment. The proposed mechanism uses a 'friendly' to the network users congestion control policy to control the transmission of the multimedia data. We implement a prototype application based on the proposed mechanism and we evaluate the proposed mechanism both in unicast and multicast transmission through a number of experiment and a number of simulations in order to examine its fairness to a group of clients and its behaviour against transport protocols (TCP) and UDP data streams. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: QoS adaptation and negotiation; performance of protocols; multicast; multimedia; QoS in heterogeneous networks

1. INTRODUCTION

Internet is a heterogeneous network environment and the network resources that are available to real-time applications can be modified very quickly. Real-time applications must have the capability to adapt their operation to network changes. In order to add adaptation characteristics to real-time applications, we can use techniques both at the network and application layers. Adaptive real-time applications have the capability to transmit multimedia data over heterogeneous networks and adapt media transmission to network changes.

Today, the underlying infrastructure of the Internet does not sufficiently support quality of service (QoS) guarantees. The new technologies, which are used for the implementation of networks, provide capabilities to support QoS in one network domain but it is not easy to

*Correspondence to: Dr Christos Bouras, Research Academic Computer Technology Institute, 61 Riga Feraiou Str., GR-26221 Patras, Greece.
† E-mail: bouras@cti.gr
‡ E-mail: gkamas@cti.gr

Received December 2001
Revised October 2002
Accepted January 2003

implement QoS among various network domains, in order to provide end-to-end QoS to the user. As a result, in the future users may have the capability to request specific end-to-end QoS even over the Internet, but this is not feasible today. In addition, many researchers take the stand that the cost for providing end-to-end QoS is too big, and it is better to invest on careful network design and careful network monitoring, in order to identify and upgrade the congested network links [1].

Multicast transmission of real-time multimedia data is an important component of many current and future emerging Internet applications, like videoconference, distance learning and video distribution. The heterogeneous nature of the Internet makes the multicast transmission of real-time multimedia data a challenge. Different clients of the same multicast multimedia data may have different processing capabilities, different loss tolerance and different bandwidth available in the paths leading to them. Should the server let the client with the least capacity dictate the adaptation? Is it fair the server ignores such a client? Generally speaking the server must treat the group of clients with fairness.

The proposed mechanism uses real-time transmission protocol/real-time control transmission protocol (RTP/RTCP) [2] for the transmission of the multimedia data. The RTP protocol seems to be the *de facto* standard for the transmission of multimedia data over the Internet and is used both by mbone tools (vit, vat, etc.) and ITU H.323 applications. In addition, RTCP offers capabilities for monitoring the transmission quality of multimedia data. We use RTCP for the implementation of the network monitoring capabilities of the proposed mechanism.

In this paper, we concentrate on the implementation of a mechanism for monitoring the network condition and estimating the appropriate transmission rate for the multimedia data. Initially, we present a unicast transmission mechanism, which we extend in order to support multicast transmission in one multicast stream. The most prominent feature of the proposed adaptive transmission mechanism is that the proposed mechanism provides the most satisfaction to the group of clients, with the current network conditions, and at the same time is trying to provide 'friendly' behaviour to other network applications. In addition, the network monitoring capabilities of the proposed mechanism is based on a combination of parameters in order to determine the network condition. Moreover, all the required modules for the implementation of the adaptive transmission mechanism are located on the server side only. This means that any application that is compatible with the transmission of multimedia data through RTP sessions (for example mbone tools) can access our service and benefit from its adaptive transmission characteristics.

This work is based on the following papers: [3, 4]. The rest of this paper is organized as follows: We present some of the related work in Section 2. In Section 3, we present the architecture of the adaptive transmission mechanism. Section 4 presents the unicast operation of the adaptive transmission mechanism and Section 5 presents the multicast operation of the adaptive transmission mechanism. In Section 6, we present the evaluation of the implemented mechanism both in unicast and multicast transmission. Finally, Section 7 concludes the paper and discusses some of our future work.

2. RELATED WORK

The subject of adaptive transmission of multimedia data over networks has engaged researchers all over the world. During the design and the implementation of an adaptive application special attention must be paid to the following critical modules: (1) the module that is responsible for the transmission of the multimedia data, (2) the module that is responsible for monitoring the network conditions and determines the change to the network conditions, (3) the module that is responsible for the adaptation of the multimedia data to the network changes, (4) the module that is responsible for handling the transmission errors during the transmission of the multimedia data. (In this paper, we are not investigating the handling of the transmission errors.)

A common approach for the implementation of adaptive applications is the use of UDP for the transmission of the multimedia data and the use of TCP for the transmission of control information [5, 6]. Another approach for the transmission of the multimedia data is the use of RTP over UDP [7, 8]. For the implementation of the network monitoring module, a common approach is the use of packet loss as an indication of congestion in the network [7, 8]. One other approach for monitoring the network conditions is the use of utilization of the client buffer [6, 9, 10]. An important factor that can be used for monitoring the network conditions, and especially for indication of network congestion, is the use of delay jitter during the transmission of the multimedia data. For the implementation of the adaptation module a common approach is the use of rate shaping [7, 11], the use of layered encoding [10], the use of frame dropping [6, 9] or a combination of the above techniques [12]. The implementation of the adaptation module depends on the encoding method that is used for the transmission of the multimedia data. For example, in order to use the frame dropping technique for the adaptation of an MPEG video stream, a selective frame dropping technique must be used, due to the fact that MPEG video uses inter-frame encoding and some frames contain information relative to other frames. Paper [13] gives a detailed survey of application level adaptation techniques.

It is important for adaptive real-time applications to have 'friendly' behaviour to the dominant transport protocols (TCP) of today's Internet [14]. Paper [15] presents a survey on TCP-friendly congestion control mechanisms.

The multicast transmission of multimedia data over the Internet has to accommodate clients with heterogeneous data reception capabilities. To accommodate heterogeneity, the server may transmit one multicast stream and determine the transmission rate that satisfies most of the clients [2, 7, 16–20, 36, 43], and may transmit multiple multicast streams with different transmission rates, and allocate clients at each stream [21, 22] or may use layered encoding and transmit each layer to a different multicast stream [7, 23–26]. An interesting survey of techniques for multicast multimedia data over the Internet is presented in paper [27].

Single multicast stream approaches have the disadvantage that clients with a low bandwidth link will always get a high-bandwidth stream if most of the other members are connected via a high bandwidth link and the same is true the other way around. This problem can be overcome with the use of a multistream multicast approach. Single multicast stream approaches have the advantages of easy encoder and decoder implementation and simple protocol operation, due to the fact that during the single multicast stream approach there is no need for synchronization of clients' actions (as the multiple multicast streams and layered encoding approaches require). The proposed single multicast stream approach can be used either autonomous or as part of a multistream multicast approach (for the management of each single stream operation). The discussion of the limitation of the single multicast stream approach (compared with multistream multicast approach) is beyond the scope of this paper.

The subject of adaptive multicast of multimedia data over networks with the use of one multicast stream has engaged researchers all over the world. Three approaches can be found in the literature for the implementation of the adaptation protocol in a single stream multicast mechanism: equation based [2, 19, 20], network feedback based [7, 16–18] or based on a combination of the above two approaches [28].

3. THE ARCHITECTURE OF ADAPTIVE TRANSMISSION MECHANISM

The architecture that we propose for the implementation of the adaptive transmission mechanism is based on the client—server model. Figure 1 shows the architecture of the adaptive transmission mechanism.

The server of the adaptive transmission mechanism consists of the following modules:

- *Video archive*: Video archive consists of a set of hard disks in which the video files are stored.
- *Feedback analysis*: This module is responsible for the analysis of feedback information from the network. The role of this module is to determine the network condition based on packet loss rate and jitter information, which are provided by RTCP receiver reports. After the examination of network condition, the feedback analysis module informs the quality adaptation module to adapt the transmission of the video to current network conditions.
- *Quality adaptation*: It is responsible for the adaptation of the video transmission quality, in order to match with the current network condition. This module is based on the rate shaping technique.
- *Packet scheduler/server buffer*: This module is responsible for the encapsulation of multimedia information in the RTP packets. In addition, this module is responsible for the transmission of the RTP packets in the network. In order to smooth accidental problems to the transmission of the multimedia data from the server to the network, we use an output buffer on the server.

The client of the adaptive transmission mechanism consists of the following modules:

- *Client buffer*: Client buffer is very important component of adaptive applications. The client stores the incoming data to the buffer before starting present data to the user. The presentation of the multimedia data to the user starts only after the necessary amount of the data is stored in the buffer. The capacity of the client buffer depends on the delay jitter during the transmission of the multimedia data. In any case, the capacity of the client

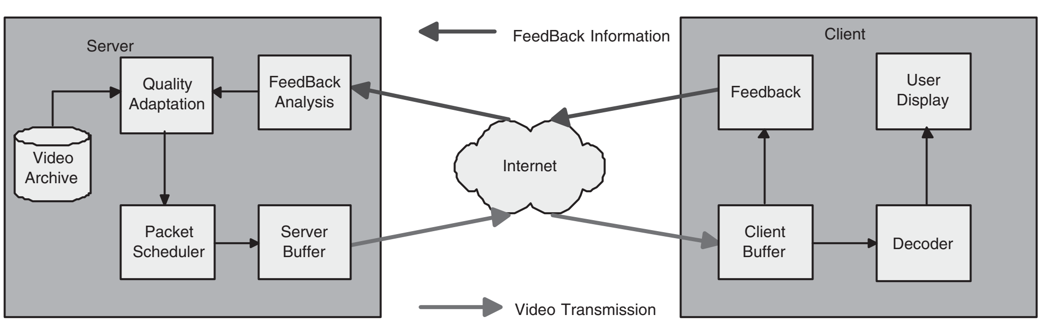


Figure 1. System architecture.

buffer must be greater than the maximum delay jitter during the transmission of the data (we measure the buffer capacity and the delay jitter in the same units, for example in seconds).

- *Feedback*: This module is responsible for monitoring the transmission quality of the data and informs the server. The monitoring of the transmission quality is based on RTCP receiver reports that the client sends to the server. RTCP receiver reports include information about the packet loss rate and the delay jitter during the transmission of the data. With the above information, the feedback analysis module of the server determines the network's condition.
- *Decoder*: This module reads the data packets from the client buffer and decodes the encoded multimedia information.
- *User display*: It is responsible for the presentation of the multimedia data to the user.

3.1. Feedback from the network

The presentation quality of real-time data depends on the packet loss rate and the delay jitter during the transmission over the network. In addition, packet losses or rapid increases of delay jitter may be considered as an indication of problems during the transmission of the data over the network. In such cases, the adaptive transmission mechanism must adapt the transmission of the data in order to avoid phenomena like network congestion.

Packet loss rate is defined as the fraction of the total transmitted packets that did not arrive at the client. In today's networks the main reason of packet losses is congestion. It is difficult to define delay jitter. Some researchers define delay jitter as the difference between the maximum and the minimum delay during the transmission of the packets for a period of time. Some other researchers define delay jitter as the maximum difference between the delay of the transmission of two sequential packets for a period of time. In this paper, in order to define the delay jitter, we use the definition that is used in RFC 1889 [29]. Delay jitter occurs when sequential packets encounter different delays in the queue of the network devices. The different delays are related to the service model of each queue and the cross-traffic in the transmission path.

The transmission host may also include delay jitter to the transmission of real-time data (host included delay jitter). This is because during the encoding of the real-time data, the encoder places a timestamp in each packet, which gives information about the time that the packet's information must be presented to the client. In addition this timestamp is used for the calculation of the delay jitter during the transmission of the real-time data. If a significant time passes from the encoding of the packet and transmission of the packet in the network (because the CPU of the transmitter host is busy), the calculation of the delay jitter is not valid. Host included delay jitter can lead to erroneous estimation for the network conditions.

As a conclusion we can say that delay jitter cannot lead to reliable estimation of network condition by itself. Delay jitter must be used in combination with other parameters, like packet loss rate, in order to make reliable estimations of the network conditions. During our experiments and simulations, we ascertain that the combination of packet loss rate and delay jitter can be used for reliable indication of network congestion.

3.2. Implementation

The implemented adaptive application supports the following video formats: MPEG, JPEG and H.263. The implementation of the above-described adaptive transmission mechanism is based on JAVA technology, and more particularly Java Media Framework (JMF) Application Programmable Interface (API). JMF provides capabilities for importing in JAVA applications and applets time-based media like video and audio. JMF supports the most common audio and video formats like AIFF, AU, AVI, GSM, MIDI, MPEG, QuickTime, RMF and WAV. In addition JMF supports transmission of real-time data (like multimedia) with the use of RTP/RTCP. Moreover, JMF gives capabilities for elaboration of audio and video data during the transmission over the network with the use of RTP/RTCP. All the above characteristics of JMF make JMF an attractive platform for the implementation of adaptive transmission mechanisms. More information about JMF can be found in Reference [30].

4. THE CASE OF UNICAST TRANSMISSION

This section describes the algorithm (feedback analysis algorithm) that is used by the adaptive transmission mechanism in order to estimate the transmission rate during the unicast transmission of the multimedia data (the feedback analysis algorithm is also used for the estimation of client *i*'s preferred transmission rate during the multicast transmission of data—more information in Section 5). We assume that we have one server that transmits multimedia data to one client. The server is using RTP/RTCP protocols for the transmission of the multimedia data. The client receives the multimedia data and informs the server for the quality of the transmission with the use of RTCP receiver reports. The server collects the RTCP receiver reports, analyses them with the use of server's feedback analysis module and determines the transmission rate that satisfies most the client with the current network conditions.

The feedback analysis module is responsible for analysing the feedback information that the client sends to the server (with the use of RTCP receiver reports), concerning the transmission quality of the multimedia data. After the analysis of the feedback information, the feedback analysis module informs the quality adaptation module to increase, decrease or keep the current transmission rate of the data. Figure 2 shows the components of the feedback analysis module.

Every time the server receives an RTCP receiver report from the client, the feedback analysis module runs the feedback analysis algorithm, which is described, in this section in order to estimate the new transmission rate, which will match the new network conditions.

The client, in repeated time spaces, sends to the server one RTCP receiver report with the use of RTCP adaptive feedback mechanism. The feedback analysis module extracts the packet loss

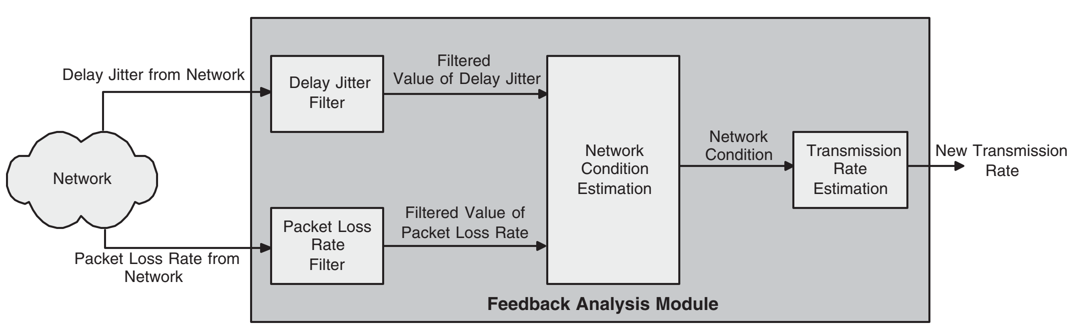


Figure 2. Feedback analysis module.

rate and the delay jitter of the RTCP receiver report and passes them through the appropriate filters (packet loss rate filter and delay jitter filter, respectively). The use of the filters is essential in order to avoid transient conditions leading to wrong estimations about the network conditions.

More particularly the value of the packet loss rate passes the following filter:

$$\mathrm{LR}_{\mathrm{new}} = a * \mathrm{LR}_{\mathrm{old}} + (1 - a) * \mathrm{LR}_{\mathrm{net}} \tag{1}$$

where:

- $\mathrm{LR}_{\mathrm{new}}$: The new filtered value of packet loss rate.
- $\mathrm{LR}_{\mathrm{old}}$: The previous filtered value of packet loss rate. When the transmission of the data starts, $\mathrm{LR}_{\mathrm{old}} = 0$.
- $\mathrm{LR}_{\mathrm{net}}$: The value of the packet loss rate from the RTCP receiver report that the client sent.
- a: This parameter specifies the feedback analysis module aggressiveness to the value of the packet loss rate, which receives from the RTCP receiver report. The parameter a satisfies $0 \leqslant a \leqslant 1$.

The value of the delay jitter passes the following filter:

$$J_{\mathrm{new}} = \beta * J_{\mathrm{old}} + (1 - \beta) * J_{\mathrm{net}} \tag{2}$$

where:

- J_{new}: The new filtered value of delay jitter.
- J_{old}: The previous filtered value of delay jitter. When the transmission of the data starts, $J_{\mathrm{old}} = 0$.
- J_{net}: The value of the delay jitter from the RTCP receiver report that the client sent.
- β: This parameter specifies the feedback analysis module aggressiveness to the value of the delay jitter, which it receives from the RTCP receiver report. The parameter β satisfies $0 \leqslant \beta \leqslant 1$. With the appropriate selection of a and β parameters values, we can set the operation of the feedback analysis module.

With the use of the above filters, we prevent a single spurious packet loss or packet delay having an excessive effect on the packet loss rate or delay jitter estimation. We have evaluated the good performance of the above filters through experiments presented in Reference [4]. Instead of the above filters, other filters, which have good smoothing behaviour, can be used. As our performance evaluation show, the above filters have good performance.

The network conditions estimation component of the network estimation module (see Figure 2) uses the filtered values of packet loss rate and delay jitter in order to characterize the network conditions. The network condition estimation component characterizes the network on the following conditions:

- *Condition congestion*: When the network is in congestion condition, the packet loss rate is high and the transmission quality of the data is low.
- *Condition load*: When the network is in load condition the transmission quality is good. The packet loss rate is in affordable value, which does not cause problems to the presentation of the multimedia data.
- *Condition unload*: When the network is in unload condition either packet losses do not exist or the packet loss rate is very small.

The changes among the network conditions are based on the filtered values of the packet loss rate and delay jitter. More particularly for the packet loss rate, we define two values $\mathrm{LR_c}$ (congestion packet loss rate) and $\mathrm{LR_u}$ (unload packet loss rate), which control the changes among the network conditions based on the following algorithm:

$$\begin{aligned}&\text{if } (\mathrm{LR_{new}} \geqslant \mathrm{LR_c}) \rightarrow \text{congestion}\\&\text{if } (\mathrm{LR_u} < \mathrm{LR_{new}} < \mathrm{LR_c}) \rightarrow \text{load}\\&\text{if } (\mathrm{LR_{new}} \leqslant \mathrm{LR_u}) \rightarrow \text{unload}\end{aligned} \tag{3}$$

No bounds exist for the filtered value of delay jitter, which designate the change among the network conditions, because the value of delay jitter depends on the transmission path and the cross-traffic in this transmission path. As a result no absolute values, which control the behaviour of delay jitter, exist. The analysis of the filtered delay jitter by the network condition estimation component is based on the fact that abrupt increases of delay jitter may denote that the buffers in the queues on the transmission path had been overloaded and this may soon cause congestion to the network. This approach ignores the host included delay jitter and we assume that the server either does not include any delay jitter to the transmission of the multimedia data or it includes a constant delay jitter during all the transmission period (in order to avoid the host included delay jitter during our experiments, we use high speed workstations as server). Network condition estimation component apprehends the abrupt increase of delay jitter as a precursor of network congestion and set the network condition to congestion. More particularly the network condition estimation component uses the following algorithm for the analysis of filtered delay jitter:

$$\text{if}(J_{\text{new}} > \gamma * J_{\text{old}}) \rightarrow \text{congestion} \tag{4}$$

where γ is a parameter specifies the network condition estimation component aggressiveness of the increase of delay jitter. In other words γ specifies quantitatively the expression 'abrupt increase of delay jitter'.

Network condition estimation component gives its estimation of the network condition to the transmission rate estimation component, which estimates the new transmission rate based on the network conditions. The transmission rate estimation component uses an additive increase, multiplicative decrease (AIMD) algorithm in order to estimate the new transmission rate. This algorithm is similar to the algorithm that the TCP rate control uses [31].

We chose an algorithm similar to TCP's rate control algorithm for fairness reasons associated with the allocation of network resources (like bandwidth) usage especially during network congestion periods. In addition, we chose an algorithm similar to TCP's rate control algorithm because, due to the fact that today's Internet does not provide end-to-end QoS, any application that sends data (mostly multimedia) over the Internet should have a friendly behaviour towards the other flows that coexist in today's Internet and especially towards the TCP flows that comprise the majority of flows [14] (the most popular applications of today's Internet (WWW–HTTP protocol, FTP, telnet, etc.) is based on TCP).

The AIMD algorithm, which we propose, is similar to the rate control algorithm of many applications of today's Internet. When the application notices available bandwidth to the network, it increases the transmission rate by adding a factor to the transmission rate (probing). In the case of network congestion, the application decreases the transmission rate by multiplying the transmission rate with a factor less than 1 (back off). More particularly, the transmission estimation module uses the following algorithm:

$$\begin{aligned} &\text{if (network = unload)} \rightarrow R_{\text{new}} = R_{\text{old}} + R_{\text{increase}} \\ &\text{if (network = load)} \rightarrow R_{\text{new}} = R_{\text{old}} \\ &\text{if (network = congestion)} \rightarrow R_{\text{new}} = R_{\text{old}} * R_{\text{decrease}} \end{aligned} \tag{5}$$

where:

- R_{new}: new value of the transmission rate,
- R_{old}: old value of the transmission rate,
- R_{increase}: factor with which the server increases the transmission rate in the case of available bandwidth,
- R_{decrease}: factor with which the server decreases the transmission rate in the case of network congestion. This factor satisfies: $0 < R_{\text{decrease}} < 1$.

The new value R_{new} of the transmission rate is used by the quality adaptation module in order to adapt the quality of the transmitted video to the new transmission rate.

The operation and the behaviour of the feedback analysis algorithm are influenced by the following parameters: $\alpha, \beta, \gamma, \text{LR}_\text{c}, \text{LR}_\text{u}, R_{\text{increase}}, R_{\text{decrease}}$. The choice of these parameters depends on the network and the kind of the dominant traffic on it. The appropriate parameters for each network can be defined through a series of experiments. In paper [7] (where a similar algorithm which is based only on packet loss rate is used) Busse *et al.* suggest specific values for some of the above parameters.

A major result of our experiments is that the choice of the above parameters ($\alpha, \beta, \gamma, \text{LR}_\text{c}, \text{LR}_\text{u}, R_{\text{increase}}, R_{\text{decrease}}$) is a trade-off. When we changed one parameter in order to improve the behaviour of the adaptive transmission mechanism to one point, the behaviour of the adaptive transmission mechanism worsened to other points. The choice of the above parameters depends on the network and especially on the dominant traffic in the network. From our performance evaluation (and other simulations and experiments, which we have performed but we did not present in this paper due to the limited space), we conclude the following values for the adaptive transmission mechanism: $a = 0.5$–0.75, $\beta = 0.8$, $\gamma = 2$, $\text{LR}_\text{u} = 0.01$–$0.02$, $\text{LR}_\text{c} = 0.05$–$0.055$, $R_{\text{increase}} = 20.000$–$50.000$, $R_{\text{decrease}} = 0.5$–$0.85$. The specific parameter values for a specific network environment can be obtained through experiments. Especially for the parameter R_{decrease} which influence significantly the behaviour of the adaptive transmission mechanism, we suggest to use a value of 0.85 in heavy congested network environments with greedy sources which do not implement any congestion control mechanism and a value of 0.5 in light congested network environments.

5. THE CASE OF MULTICAST TRANSMISSION

This section describes the multicast operation of the adaptive transmission mechanism. We assume that we have a server, which transmits multimedia data to a group of n clients with the use of multicast in one stream. The server is using RTP/RTCP protocols for the transmission of the multimedia data. Clients receive the multimedia data and inform the server of the quality of the transmission with the use of RTCP receiver reports. The estimation of transmission rate that satisfies most of the group of clients is done with the use of multicast transmission rate

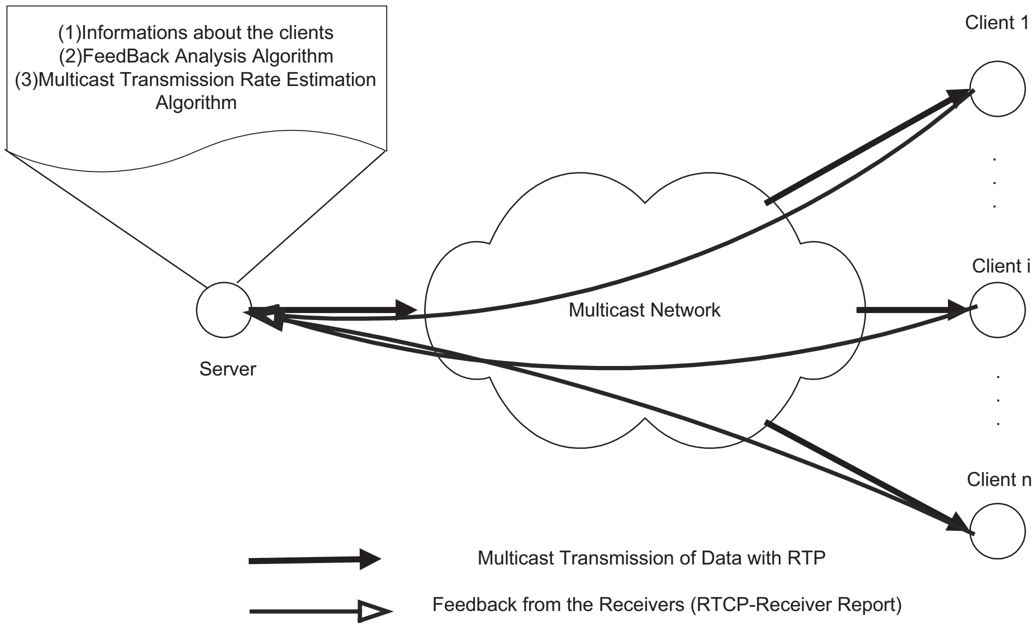


Figure 3. Multicast operation of the adaptive transmission mechanism.

estimation algorithm which is presented in this section. Figure 3 shows the multicast operation of the adaptive transmission mechanism.

The server keeps information about each client i, and each time receives one RTCP receiver report from client i, estimates the client i's preferred transmission rate r_i (which represent the transmission rate that this client will prefer if it was the only one client in the multicast transmission of the multimedia data). The estimation of client i's preferred transmission rate r_i is done with the use of feedback analysis algorithm, which is described in Section 4. If the server transmission rate r exceeds the client i's preferred transmission rate r_i, then this client experiences performance degradation due to packet losses. If the server transmission rate r is less than the client i's preferred transmission rate r_i, then this client experiences performance degradation due to unutilized bandwidth. We define a client fairness function RF that maps from the actual operating rate to a fairness value. For client i the fairness function RF_i must satisfy the following constraints: (1) $\mathrm{RF}_i(r) \in [0.0, 1.0]$, the fairness is normalized to a quantity between 0.0 and 1.0, (2) $\mathrm{RF}_i(r_i) = 1$, where r_i is the client i's preferred transmission rate, (3) $\mathrm{RF}_i(r) < 1.0$ if $r \neq r_i$, which means that RF_i has a single maximum $r = r_i$, (4) RF_i is non-decreasing in the range $[0, r_i]$, which means that as the throughput increases the fairness remains constant or increases and (5) RF_i is non-increasing in the range $[r_i, \infty]$, which means that as the losses increase, the fairness remains constant of decreases.

The server uses the IRF and RF_i functions which are presented in Reference [18], in order to determine the transmission rate that satisfy most of the group of clients. RF_i function for the client i is defined in Reference [18] as follows and meets the above-described requirements for the receive fairness function:

$$\mathrm{RF}_i(r) = \frac{\min(r_i, r)}{\max(r_i, r)} \tag{6}$$

where r_i is the transmission rate that the client i prefers and r is the transmission rate that the server is planning to use. From Equation (6) it is obvious that the RF_i function has values in [0.0, 1.0], and the client i is satisfied when the $\mathrm{RF}_i \approx 1.0$ and completely satisfied when $\mathrm{RF}_i = 1.0$ (when $r_i = r$). The client i is not satisfied when the $\mathrm{RF}_i \ll 1.0$. Client i can encounter dissatisfaction either of packet loss (when $r_i < r$) or of unutilized bandwidth (when $r_i > r$). IRF function for a group of n clients is defined in Reference [18] as follows:

$$\mathrm{IRF}(r) = \sum_{i=1}^{n} a_i * \mathrm{RF}_i(r) \tag{7}$$

subject to $\sum_{i=1}^{n} a_i = 1$ and $0 \leqslant a_i \leqslant 1, 0,\ i = 1, \ldots, n$, where r is the transmission rate that the server is planning to use and a_i is the weight of the client i to the computation of the IRF value. From Equation (7), it is obvious that for greater values of IRF function, the group of clients is more satisfied and for lesser values of IRF function, the group of clients is less satisfied.

The server in repeated time spaces estimates the transmission rate r for the multicast transmission of the multimedia data. The server is using as a satisfaction measurement the IRF function defined in Equation (7), and is usually treating all clients as equal, which means that the weight a_i for all the clients $i, i = 1 \ldots n$ in IRF function is $a_i = 1/n$.[§] If the server wants to treat unequally the group of clients, it can assign priority to some clients with the use of unequal a_i values.

The server in repeated time spaces estimates the transmission rate r for multicast multimedia data with the use of multicast transmission rate estimation algorithm. The estimation of the server transmission rate r is aiming to increase the satisfaction of the group of clients based on the satisfaction measurement that the function IRF of Equation (7) provides.

The server is using an AIMD mechanism in order to estimate the new transmission rate r. This algorithm is similar to the algorithm that the TCP rate control uses. We chose an algorithm similar to TCP's rate control algorithm for fairness reasons to the allocation of network resources (like bandwidth), especially during network congestion periods.

When the server is estimating the new transmission rate r, it has three opportunities: (1) To increase the transmission rate by adding a factor, R_{increase} (r_{incr}), (2) to keep the previous transmission rate (r_{stay}) and (3) to decrease the transmission rate by multiplying with a factor less than 1, R_{decrease} (r_{dcr}).

The multicast transmission rate estimation algorithm is selecting as new transmission rate r, the transmission rate r from $\{r_{\mathrm{incr}}, r_{\mathrm{stay}}, r_{\mathrm{dcr}}\}$ which provides the most satisfaction to the group of clients, which means the transmission rate r from $\{r_{\mathrm{incr}}, r_{\mathrm{stay}}, r_{\mathrm{dcr}}\}$ that has the greater IRF value. In addition the multicast transmission rate estimation algorithm is updating the old value of the preferred transmission rates of all the clients in order that the feedback analysis algorithm should be aware of the current transmission rate. Here is the summary of the multicast transmission rate estimation algorithm operation:

$$r_{\mathrm{incr}} = r_{\mathrm{old}} + R_{\mathrm{increase}} \tag{8}$$

[§]The number n of the receivers can easily be computed by the RTCP protocol.

$$r_{\text{stay}} = r_{\text{old}}$$

$$r_{\text{dcr}} = r_{\text{old}} * R_{\text{decrease}}$$

$$r_{\text{new}} = \max \text{IFR}_{r=r_{\text{incr}}, r_{\text{stay}}, r_{\text{dcr}}}[\text{IFR}(r)]$$

$$\text{receiver} - i_{i=1\ldots n}\text{: } R_{i\text{-old}} = r_{\text{new}}$$

$$r_{\text{old}} = r_{\text{new}}$$

where r_{new} is the new transmission rate of the server, and r_{old} is the previous transmission rate of the server. In addition, the transmission rate r_{new} cannot be greater than a value $r_{\max}$ and cannot be smaller than a value $r_{\min}$. The values of $r_{\max}$ and $r_{\min}$ depend on the network and application type (e.g. video encoding, server network connection, etc.).

The multicast transmission rate estimation algorithm does not take directly into account the current network condition, during the estimation of new transmission rate r_{new} for the server. The current network conditions are taken directly into account by the feedback analysis algorithm, during the estimation of clients' preferred transmission rate r_i. Because the values of the clients' preferred transmission rates r_i are involved is the calculation of IFR(r), the multicast transmission rate estimation algorithm takes indirectly into account the current network conditions. Our simulation experiments (Section 6) show that the approach of multicast transmission rate estimation algorithm to take in account directly the satisfaction of the clients directly and indirectly the current network condition and works well.

6. PERFORMANCE EVALUATION

6.1. Unicast transmission

In this section, we present a number of experiments that we made in order to analyse the behaviour of the adaptive transmission mechanism during unicast transmission and prove the concept of the proposed mechanism. The primary aim of the experiments was the study of the adaptive transmission mechanism behaviour regarding the dominant traffic model of today's Internet. Experiments realization to the Internet is a difficult procedure due to the fact that the Internet is a very changeable environment, and it is difficult to achieve the same fixed conditions in every experiment. In the Internet, it is possible that accidental events affect the results of the experiments.

For the above reason, and in order to realize our experiments to a stable network environment, we used the following network environment as testbed: With the use of ATM infrastructure, we established a virtual circuit (VC), with CBR characteristics, between the server and the client of the adaptive transmission mechanism. The VC had capacity 300 kbps and at the same time we transmitted video with the use of adaptive transmission mechanism and TCP or UDP traffic in order to evaluate the behaviour of the adaptive transmission mechanism.

The video file, which was transmitted through the VC, was encoded to ITU-H.263 encoding format and the encoder initially was producing data with a rate of 50 kbps. The video file had 5 min duration and dimension $177 * 144$ pixels (CIF format).

We chose for the realization of our experiments to transmit low-quality video in order to ensure that the delay jitter which was appearing during the experiments was originated by the network and not by the server (host included delay jitter). We performed two experiments:

- *Unicast transmission of adaptive video and UDP traffic at the same time*: During this experiment, we investigated the behaviour of the adaptive transmission mechanism during heavy congestion produced by greedy UDP traffic.
- *Unicast transmission of adaptive video and TCP traffic at the same time*: During this experiment, we investigated the behaviour of the adaptive transmission mechanism against TCP traffic.

6.1.1. Experiment one: unicast transmission of adaptive video and UDP traffic. During this experiment, we used the following scenario: initially, we transmitted only video to the VC with the use of adaptive transmission mechanism. Two minutes after the beginning of video transmission, we transmitted for the following 2 min, together with the adaptive video, UDP traffic through the VC. After the end of UDP traffic, the transmission of adaptive video continued for 1 min until the video files ended. We select this scenario in order to investigate the proposed mechanism behaviour during heavy congestion condition produced by greedy UDP traffic, which does not implement any congestion control mechanism. Other topologies and scenarios that ensure heavy congestion condition can also be used.

Figures 4 and 5 show the video transmission rate, the packet loss rate and the delay jitter during experiment one. In all charts, in this section the video transmission rate refers only to video data (payload) and does not include headers data. The header overhead (RTP header, UDP header, IP header and ATM cell header) during our experiments was significant. For this reason, the video transmission rate never reached all the capacity of the VC (300 kbps). During this experiment the parameters which control the behaviour of the adaptive transmission mechanism had the following values $\alpha = 0.5$, $\beta = 0.8, \gamma = 2$, $LR_c = 0.05$, $LR_u = 0.02$, $R_{increase} = 20.000$(bps), $R_{decrease} = 0.85$ (we obtain these values after a number of experiments in the described topology). Initially, the adaptive transmission mechanism started to transmit video with transmission rate 50 kbps and gradually reserved all the available bandwidth in the VC.

In minute two, when the transmission of UDP traffic started, congestion occurred to the network, and the adaptive transmission mechanism perceived the congestion condition through the increase of packet loss rate and delay jitter. Due to this congestion condition, the adaptive

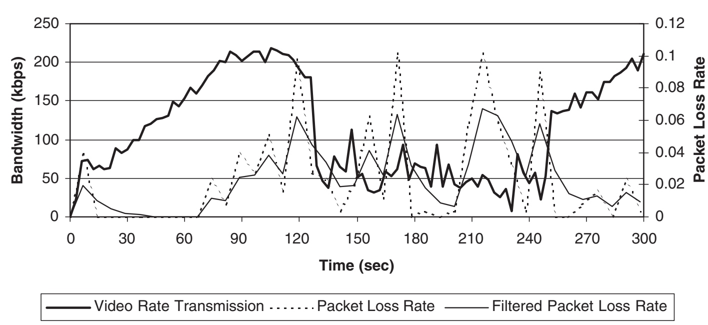


Figure 4. Video transmission rate and packet loss rate during experiment one.

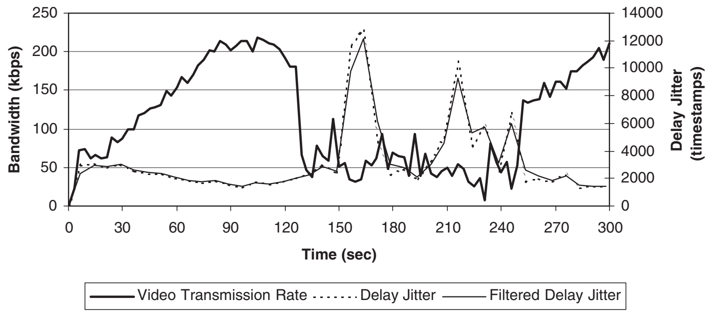


Figure 5. Video transmission rate and delay jitter during experiment one.

transmission mechanism reduced its transmission rate near to 50% and kept this transmission rate for the next 2 min, during which the transmission of UDP traffic took place. When the transmission of UDP traffic stopped (minute 4) the adaptive transmission mechanism gradually reserved again all the available bandwidth.

From Figures 4 and 5, it is obvious that the adaptive transmission mechanism has good behaviour during heavy congestion condition. It is important for all network users to have 'friendly' behaviour to other users for the proper operation of a network. The fact that the transmission rate of the adaptive transmission mechanism was dropped beyond 50% of the total available bandwidth during this experiment, was because UDP traffic (that we use during our experiment) did not have any congestion control policy. In addition, Figure 5 shows how important is the use of delay jitter as network congestion indication. During, this experiments, some times the packet loss rate was indicating load network ($0.02 < \text{packet loss rate} < 0.05$) but the rapid increase of delay jitter was indicating correctly network congestion.

6.1.2. Experiment two: unicast transmission of adaptive video and TCP traffic. During this experiment, we used the following scenario: Initially, we transmitted only video to the VC with the use of the adaptive transmission mechanism. Two minutes after the beginning of video transmission, we transmitted, together with the adaptive video, TCP traffic through the VC. After the end of TCP traffic the transmission of adaptive video continued for 1 min until the video files ended. When the video transmission started, the adaptive transmission mechanism had a transmission rate of 50 kbps. We select this scenario in order to investigate the proposed mechanism behaviour during congestion condition produced by TCP traffic, which implements the well known AIMD congestion control mechanism. Other topologies and scenarios that ensure congestion condition can also be used.

Initially, we used the same values as in the first experiment for the parameters that control the behaviour of adaptive transmission mechanism ($\alpha = 0.5$, $\beta = 0.8$, $\gamma = 2$, $\mathrm{LR_c} = 0.05$, $\mathrm{LR_u} = 0.02$, $R_{\mathrm{increase}} = 20.000$ (bps), $R_{\mathrm{decrease}} = 0.85$) (we obtain these values after a number of experiments in the described topology). Figures 6 and 7 show the video transmission rate, the packet loss rate and the delay jitter during this experiment.

As Figures 6 and 7 show, with the above parameters the adaptive transmission mechanism does not have 'friendly' behaviour to TCP traffic. When the transmission of TCP traffic started, the adaptive transmission mechanism released only a small amount of bandwidth, which was re-allocated in few seconds. The reason for the above behaviour was the different congestion

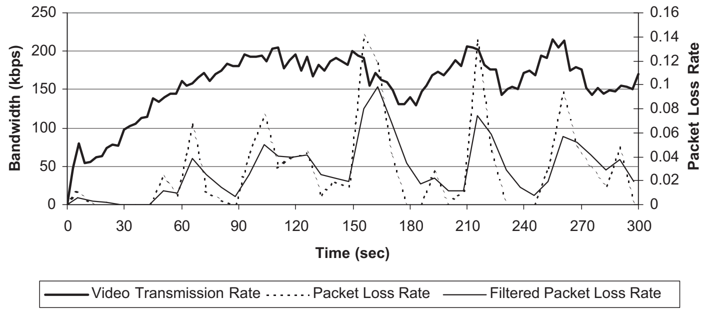


Figure 6. Video transmission rate and packet loss rate during experiment two with $R_{\text{decrease}} = 0.85$.

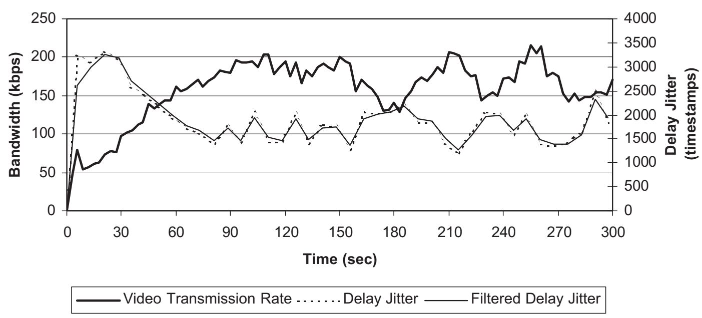


Figure 7. Video transmission rate and delay jitter during experiment two with $R_{\text{decrease}} = 0.85$.

control policies of TCP traffic and adaptive transmission mechanism. In the case of network congestion, the adaptive transmission mechanism decreased its transmission rate to 85% ($R_{\text{descrease}} = 0.85$) and the TCP traffic decreased its transmission rate to 50%. The more aggressive behaviour of adaptive transmission mechanism in congestion control resulted in the TCP traffic being able to allocate enough bandwidth. This behaviour of adaptive transmission mechanism is not desirable because TCP traffic is one of the dominant traffics of today's Internet. In order to adjust the behaviour of adaptive transmission mechanism to be 'friendly' with TCP, we repeated the above experiment and we set the parameter $R_{\text{decrease}} = 0.5$. With this change, the congestion control policy of the adaptive transmission mechanism becomes 'friendly' to TCP congestion control policy. Figures 8 and 9 show the video transmission rate, the packet loss rate and the delay jitter during the repeat of the experiment.

As Figures 8 and 9 show, with the change of parameter R_{decrease} the behaviour of adaptive transmission mechanism to TCP traffic became better. When we started to transmit TCP traffic and network congestion occurs, the adaptive transmission mechanism released bandwidth in order to be used by the TCP traffic. Consecutively, the adaptive transmission mechanism kept steady its transmission rate until the transmission of TCP stopped. Then the adaptive transmission mechanism gradually reserved again all the available bandwidth. The better behaviour of the adaptive transmission mechanism was because adaptive transmission mechanism and TCP traffic followed the same congestion control policy: when congestion

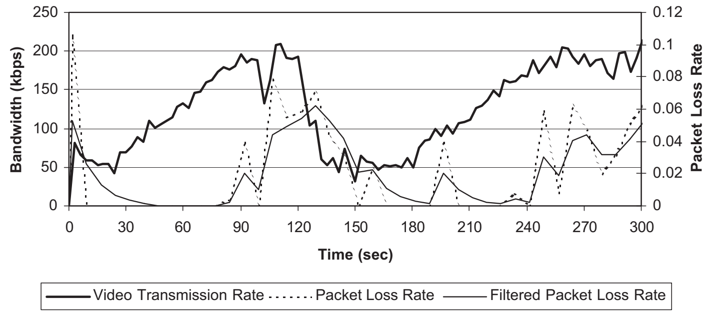


Figure 8. Video transmission rate and packet loss rate during experiment two with $R_{\text{decrease}} = 0.5$.

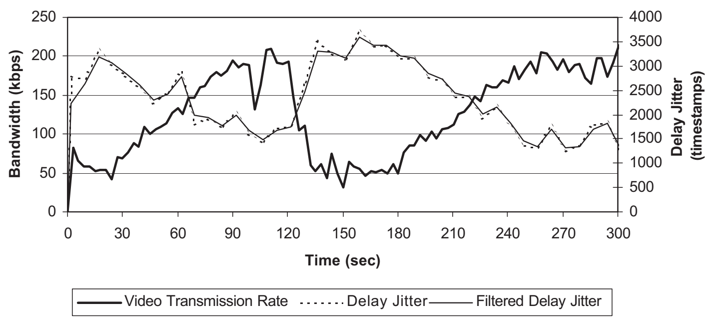


Figure 9. Video transmission rate and delay jitter during experiment two with $R_{\text{decrease}} = 0.5$.

occurred, they decreased the transmission rate to 50%. From Figures 8 and 9, it is obvious that both packet loss rate and delay jitter indicates network congestion.

6.2. *Multicast transmission*

In this section, we present a number of simulations that we made in order to analyse the behaviour of the adaptive transmission mechanism during the multicast transmission of multimedia data with the use of one stream and prove the concept of the proposed mechanism. We implemented our mechanism and ran simulations in the LBNL network simulator ns-2 [32].

Ns-2 is a discrete event simulator targeted at networking research. Ns-2 provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Ns-2 provides a split-level programming model in which packet processing is done in a systems language while simulation set-up is done in a scripting language. In addition ns-2 with the use of Nam tool [33], supports simple scenario editing and provides visualization output. We choose the ns-2 network simulator for the following reasons: (1) ns-2 provides a network protocol library which support the most of today's Internet protocols, (2) ns-2 network protocol library supports RTP/RTCP protocol in which our implementation is based and (3) ns-2 is a free tool with a big user community and many documentation (manuals, tutorials, discussion lists, etc.) available.

We chose to evaluate the multicast operation of the adaptive transmission mechanism through simulation because it is not easy to implement controlled network testbeds with many users in today's Internet. In addition, we have the opportunity to compare the operation of the adaptive transmission mechanism in a network environment and in a simulation environment. As Section 6 shows, the operation of the adaptive transmission mechanism is similar both to the network environment and the simulation environment.

We ran three simulations: (1) multicast transmission of adaptive multimedia in heterogeneous clients, (2) multicast transmission of adaptive multimedia in heterogeneous clients and UDP traffic at the same time and (3) multicast transmission of adaptive multimedia in heterogeneous clients and TCP traffic at the same time. During all the simulations we use the following values for the parameters of our algorithms: $\alpha = 0.75$, $\beta = 0.8$, $\gamma = 2$, $\mathrm{LR_c} = 0.055$, $\mathrm{LR_u} = 0.01$, $r_{\max} = 2.000.000$ bps, $r_{\min} = 200.000$ bps, $R_{\text{increase}} = 50.000$ bps, $R_{\text{decrease}} = 0.5$ and $a_i = 1/n,\ i,$ $i = 1 \ldots n$, where n is the number of the clients (we obtain these values after a number of experiments in the described topology). During our simulations we had 20 clients.

6.2.1. Simulation one: multicast transmission of adaptive multimedia in heterogeneous clients. Figure 10 shows the topology of this simulation. The bandwidth of each link is given to the simulation topology and varies from 0.5 to 2.0 Mbps. All the links in the simulation topology had a delay of 10 ms, was full duplex and they used the drop-tail¶ (FIFO) [34] policy to their queue. Tail-drop is a means of avoiding congestion that treats all traffic equally and does not differentiate between classes of service. Queues fill during periods of congestion. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full. During this simulation, we had one server (S) that multicast multimedia data to a group of 20 clients (C1–C20) with the use of the adaptive transmission mechanism. Clients C1–C10 were connected to router n2 and clients C11 to C20 were connected to router n3. The clients transmitted RTCP receiver reports with the use of the RTCP adaptive feedback mechanism and the server ran the multicast transmission rate estimation algorithm every 1 s. We ran this simulation for 100 s and the server started transmitting data with transmission rate of 1.5 Mbps. We select this scenario in order to investigate the fairness of the adaptive transmission mechanism in a topology where the excepted behaviour (fairness against a heterogeneous group of receivers) can easily be identified. Other topologies and scenarios where the excepted behaviour can easily be identified can also used.

Figure 11 shows the server transmission rate and the values of the IRF function. When the server started transmitting data with a transmission rate of 1.5 Mbps all the clients, except C1, C2, C3, C11 and C12, encountered dissatisfaction due to packet losses because their available bandwidth was less than 1.5 Mbps. The server started reducing the transmission rate in order to treat with fairness all the clients. The server reduced its transmission rate near to 0.6 Mbps (5th second). At this point, the dissatisfaction that the 'fast' clients (for example C1 or C9) encountered due to unutilized bandwidth was more than the dissatisfaction that the 'slow' (for example C4 or C12) clients encountered due to packet losses. The server started increasing the transmission rate in order to treat with fairness all the clients. At the 15th second the transmission rate of the server was stabilized near 1.0 Mbps and the server maintained this

¶Drop-tail is the most common queue policy to Internet routers.

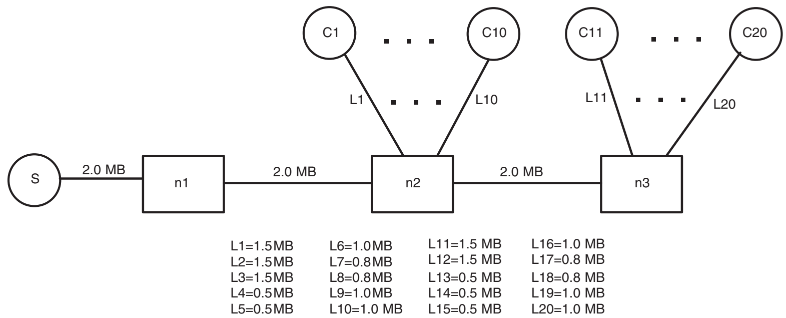


Figure 10. Topology of simulation one.

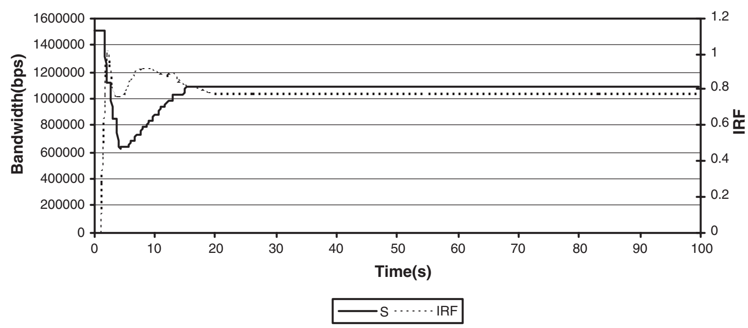


Figure 11. Server bandwidth and IRF function values of simulation one.

transmission rate until the end of the simulation. At the 15th second the server had found the transmission rate that satisfied most of the group of clients with the current network conditions. In addition from 15th to 100th second the value of IRF function was stable because the server did not change its transmission rate.

The adaptive transmission mechanism behaved well: after some time the server found the transmission rate that satisfied most the group of clients and kept that transmission rate while the network conditions were not changed. In addition, the value of transmission rate ($\sim$ 1.0 Mbps) that satisfied most the group of clients was the expected due to the fact that the most of the clients preferred transmission rate of 1.0 Mbps.

6.2.2. Simulation two: multicast transmission of adaptive multimedia in heterogeneous clients and UDP traffic at the same time. Figure 12 shows the topology of this simulation, which is the same as the topology of simulation one, except for that we had added two nodes A and B connected to router n1 and router n3, respectively. We had again one server (S) that multicast multimedia data to a group of 20 clients (C1–C20) with the use of the adaptive transmission mechanism. Clients C1–C10 were connected to router n2 and clients C11–C20 were connected to router n3. In order to produce UDP traffic, we attached to node A, a constant bit rate (CBR) traffic generator (CBR-source), which transmitted data to a CBR-receiver attached to node B. The CBR-source produced UDP traffic with constant transmission rate of 1.5 Mbps. The clients

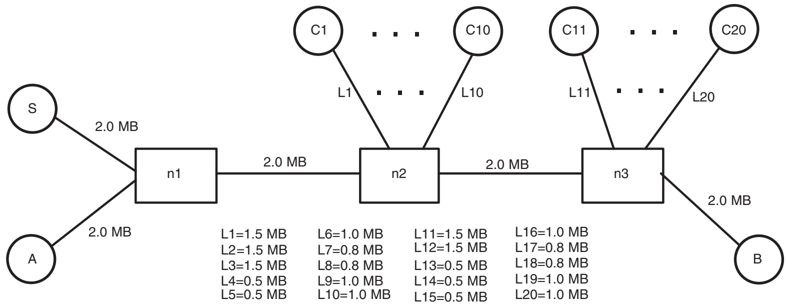


Figure 12. Topology of simulations two.

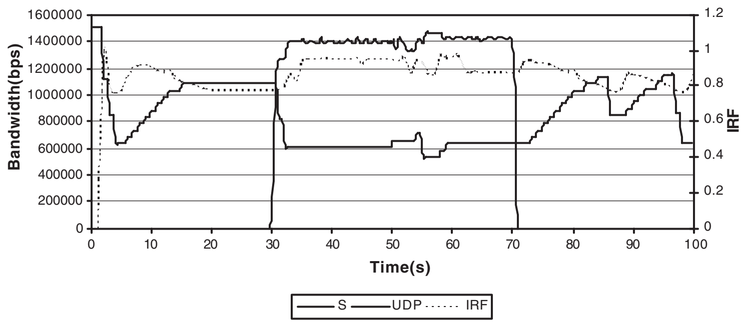


Figure 13. Server bandwidth, UDP-receiver bandwidth and IRF function values of simulation two.

transmitted RTCP receiver reports with the use of the RTCP adaptive feedback mechanism and the server ran the multicast transmission rate estimation algorithm every 1 s. We ran this simulation for 100 s and the server started transmitting data with transmission rate of 1.5 Mbps. The CBR-source started the transmission of the data at the 30th second, and stopped the transmission of the data at the 70th second. We select this scenario in order to investigate the proposed mechanism behaviour during heavy congestion condition produced by greedy UDP traffic, which does not implement any congestion control mechanism.

Figure 13 shows the server transmission rate, the CBR-receiver bandwidth and the values of IRF function. The server found the transmission rate that satisfied most the group of clients (15th second) after some initial instability. When the transmission of UDP traffic started (at 30th second), congestion occurs to links between the router n1, n2 and between router n2, n3. The clients preferred smaller transmission rates due to the congestion condition, and the server reduced its transmission rate near 0.5 Mbps and kept this transmission rate for the next 40 s, during which time the transmission of UDP traffic took place. When the transmission of UDP traffic stopped (70th second), the server gradually reserved again the available bandwidth. The value of IRF function was stable when the transmission rate of the server was stable, and floated between 0.77 and 0.97 when the transmission of UDP traffic took place. The IRF function had higher values, when the transmission of the UDP traffic took place, because all the clients encountered packet losses due to congested links between the router n1, n2 and between router n2, n3 and all the clients were satisfied with the small transmission rate that the server selected.

It is obvious from Figure 13 that the proposed mechanism has good behaviour during network congestion condition. When the transmission of UDP traffic started, the server reduced its transmission rate and when the transmission of UDP traffic stopped, the server reserved again the available bandwidth.

6.2.3. Simulation three: multicast transmission of adaptive multimedia in heterogeneous clients and TCP traffic at the same time. Figure 14 shows the topology of this simulation, which is the same as the topology of simulation two except that the capacity of some links has changed. We had again one server (S) that multicast multimedia data to a group of 20 clients (C1–C20) with the use of the adaptive transmission mechanism. Clients C1–C10 were connected to router n2 and clients C11–C20 were connected to router n3. In order to produce TCP traffic, we connected to node A and B, an FTP server and an FTP client, respectively. The FTP server transmitted a file to FTP client using '4.3BSD Tahoe TCP' protocol [31]. The clients transmitted RTCP receiver reports with the use of the RTCP adaptive feedback mechanism and the server ran the multicast transmission rate estimation algorithm for every 1 s. We ran this simulation for 100 s and the server started transmitting data with transmission rate of 1.5 Mbps. The transmission of the file from the FTP server to FTP client, started at the 30th second and stopped at the 70th second. We select this scenario in order to investigate the proposed mechanism behaviour during congestion condition produced by TCP traffic, which implements the well-known AIMD congestion control mechanism. Other topologies and scenarios that ensure congestion condition produced by TCP traffic can also be used.

Figure 15 shows the server transmission rate, the TCP source bandwidth and the values of IRF function. The server found the transmission rate that satisfied most of the group of clients (15th second) after some initial instability. When the transmission of TCP source started (at 30th second), congestion occurs to links between the router n1, n2 and between router n2, n3. The clients preferred smaller transmission rates due to congestion condition, and the server released bandwidth in order the TCP traffic to use it. When the transmission of the TCP traffic took place, the server realized some bandwidth (about 0.3 Mbps) for a while and reserved it again. When the transmission of TCP traffic stopped (70th second) the server gradually reserved again the available bandwidth. The value of IRF function was stable when the transmission rate of the server was stable and floated between 0.79 and 0.90 when the transmission of TCP traffic

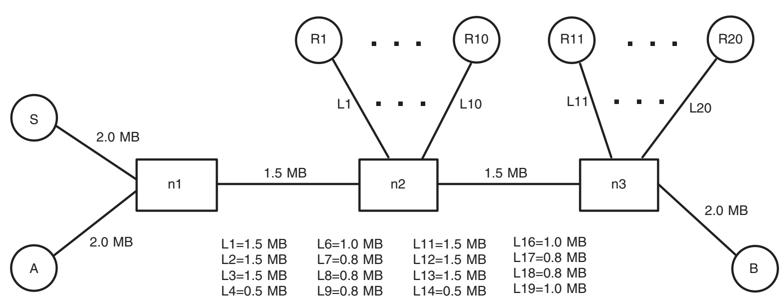


Figure 14. Topology of simulations three.

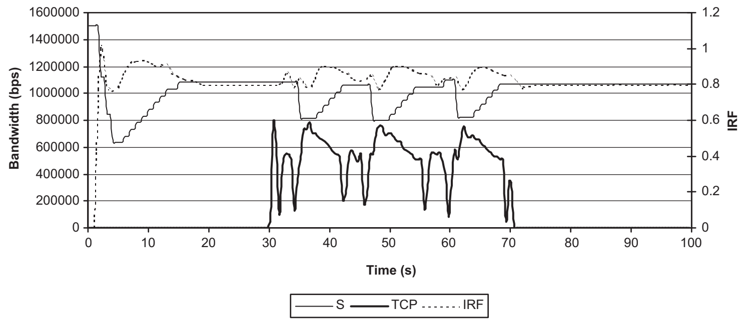


Figure 15. Server bandwidth, TCP source bandwidth and IRF function values of simulation three.

took place, because the transmission of TCP traffic produced instability to the adaptation mechanism and the server changed continually its transmission rate.

It is obvious from Figure 15 that the behaviour mechanism to TCP traffic is 'friendly'. The TCP traffic had transmission rate of more than 0.5 Mbps many times and maximum transmission rate of 0.8 Mbps during the simulation, which is good performance for TCP transmission. In addition, the server many times realized bandwidth and provided it to TCP source and in one case (32nd second) the server realized 0.3 Mbps of its bandwidth. The proposed mechanism had the following drawback: the server's transmission rate during the transmission of TCP traffic was not stable. The server would have ideal behaviour if it reduced its transmission rate and kept it steady while the transmission of TCP traffic took place.

7. CONCLUSION—FUTURE WORK

In this paper, we present a mechanism for unicast and multicast transmission of adaptive multimedia data in a heterogeneous group of clients. The proposed mechanism uses RTP/RTCP protocols for the transmission of multimedia data. We are concentrating on the design of a mechanism for monitoring the network condition and estimating the appropriate rate both for unicast and multicast transmission of multimedia data in order to treat with fairness the clients. In addition, we investigated the behaviour of the proposed mechanism against the dominant transport protocols of today's Internet (TCP and UDP) through a number of experiments and a number of simulations. Through the above experiment and simulations, we draw the following conclusions: (1) the proposed mechanism treats with fairness the group of clients, (2) the proposed mechanism has 'friendly' behaviour to TCP traffic streams and (3) the proposed mechanism behaves well during heavy congestion conditions.

Our future work includes the improvement of the proposed mechanism's behaviour against TCP traffic. In addition, we will investigate the behaviour of the proposed mechanism during the multicast transmission in very large group of clients. The multicast transmission to a very large group of clients encounters the feedback implosion problem [16]. Furthermore, we will investigate the scalability of proposed mechanism and how the proposed mechanism will deal with the feedback implosion problem. Moreover, we plan to extend the proposed mechanism with the use of multicast in multiple streams in order to treat with more fairness a heterogeneous group of clients. Finally we intend to enhance the implementation by adding a mechanism in order to dynamically choose and modify the parameters that regulate the aggressiveness of the adaptation.

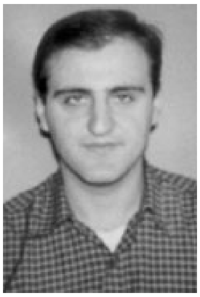
REFERENCES

1. Diot C. On QoS & Traffic Engineering and SLS-related Work by Sprint. *Workshop on Internet Design for SLS Delivery*, Tulip Inn Tropen, Amsterdam, The Netherlands, 25–26 January 2001.
2. Rizzo L. pgmcc: A TCP-friendly single-rate multicast congestion control scheme. *Proceedings of SIGCOMM'2000*, Stockholm, August 2000.
3. Bouras Ch, Gkamas A. A mechanism for multicast multimedia data with adaptive QoS characteristics. *Protocols for Multimedia Systems*, Enschede, The Netherlands, 17–19 October, 2001.
4. Bouras Ch, Gkamas A. Streaming multimedia data with adaptive QoS characteristics. *Protocols for Multimedia Systems*, Cracow, Poland, 22–25 October, 2000; 129–139.
5. Cowan C, Cen S, Walpole J, Pu C. Adaptive methods for distributed video presentation. *ACM Computing Surveys* 1995; **27**(4):580–583 (Symposium on Multimedia).
6. Walpole J, Koster R, Cen S, Cowan C, Maier D, McNamee D, Pu C, Steere D, Yu L. A player for adaptive mpeg video streaming over the internet. In *Proceedings of the 26th Applied Imagery Pattern Recognition Workshop AIPR-97*. SPIE: Washington, DC, October 1997.
7. Byers J, Frumin M, Horn G, Luby M, Mitzenmacher M, Roetter A, Shaver W. FLID-DL: congestion control for layered multicast. In *Proceedings of NGC 2000*, November 2000; 71–81.
8. Jacobs S, Eleftheriadis A. Real-time video on the web using dynamic rate shaping. Department of Electrical Engineering, Columbia University, New York, NY 10027.
9. Hess CK. Media streaming protocol: an adaptive protocol for the delivery of audio and video over the internet. *Master's Thesis*, University of Illinois, Urbana-Champaign, 1998.
10. Rejaie R, Estrin D, Handley M. Quality adaptation for congestion controlled video playback over the internet. In *Proceedings of ACM SIGCOMM '99*, Cambridge, September 1999.
11. Jacobs S, Eleftheriadis A. Adaptive video applications for non-QoS networks. *Proceedings of the 5th International Workshop on Quality of Service* (*IWQoS'97*), Columbia University, New York, USA, 1997; 161–165.
12. Ramanujan R, Newhouse J, Kaddoura M, Ahamad A, Chartier E, Thurber K. Adaptive streaming of MPEG video over IP networks. *Proceedings of the 22nd IEEE Conference on Computer Networks* (LCN'97), November 1997.
13. Vandalore B, Feng W, Jain R, Fahmy S. A survey of application layer techniques for adaptive streaming of multimedia. *Real Time Imaging* 2001; **7**(3):221–235.
14. Floyd S, Fall K. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking* 1999; **7**(4):458–472.
15. Widmer J, Denda R, Mauve M. A survey on TCP-friendly congestion control mechanisms. *Special Issue of the IEEE Network Magazine*, *Control of Best Effort Traffic*, 2001.
16. Bolot JC, Toilette T, Wakeman I. Scalable feedback control for multicast video distribution in the internet. In *Proceedings of SIGCOMM 1994*, ACM SIGCOMM, London, England, August 1994; 139–146.
17. Sisalem D. Fairness of adaptive multimedia applications. *1998 IEEE International Conference on Communications. Conference Record* (Affiliated with SUPERCOMM'98 IEEE), 1998; 791–796.
18. Jiang T, Ammar W, Zegura EM. Inter-receiver fairness: a novel performance measure for multicast ABR sessions. *SIGMETRICS* 1998; 202–211.
19. Pandhye J, Kurose J, Towsley D, Koodli R. A model based TCP-friendly rate control protocol. *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video* (*NOSSDAV*), Basking Ridge, NJ, June 1999.
20. Widmer J, Handley M. Extending equation-based congestion control to multicast applications. *Proceedings of ACM SIGCOMM*, San Diego, CA, August 2001.
21. Jiang T, Zegura EW, Ammar M. Inter-receiver fair multicast communication over the internet. In *Proceedings of the 9th International Workshop on Network and Operating Systems Support for Digital Audio and Video* (*NOSSDAV*), June 1999; 103–114.
22. Cheung SY, Ammar M, Xue L. On the use of destination set grouping to improve fairness in multicast video distribution. In *Proceedings of INFOCOM 96*, March 1996, San Fransisco, Georgia tech, College of Computing, *Technical Report GIT-CC-95-25*, July 1995.
23. McCanne S, Jacobson V. Receiver-driven layered multicast. *1996 ACM Sigcomm Conference*, August 1996; 117–130.

25. Vickers BJ, Albuquerque CVN, Suda T. Adaptive multicast of multi-layered video: rate-based and creditbased approaches. *Proceedings of IEEE Infocom*, March 1998.
26. Tsaoussidis V, Wei S. *Reliability/Throughput/Jitter Tradeoffs For Real-time Transport Protocols*. Department of Computer Science.
27. Li X, Ammar MH, Paul S. *Video Multicast over the Internet IEEE Network Magazine*, April 1999.
28. Sisalem D, Wolisz A. LDA + TCP-friendly adaptation: a measurement and comparison study. In *10th International Workshop on Network and Operating Systems Support for Digital Audio and Video* (*NOSSDAV'2000*), Chapel Hill, NC, USA, 25–28 June, 2000.
29. Shculzrinne H, Casner, Frederick, Jacobson. RTP: A transport protocol for real-time applications. *RFC 1889*, IETF, January 1996.
30. Java Media Framework: http://java.sun.com/products/java-media/jmf/index.html
31. Stevens W. TCP Slow start, congestion avoidance, fast retransmit and fast recovery algorithms. *RFC 2001*, January 1997.
32. McCanne S, Floyd S. The UCB/LBNL network simulator, software online. http://www.isi.edu/nsnam/ns/
33. Estrin D *et al*. Network visualization with the VINT network animator, Nam. *Technical Report* 99-703, Department of Computer Science, University of Southern California, Los Angeles, 1999.
34. First-in, First-out (FIFO) Queuing: http://www.juniper.net/techcenter/techpapers/200020-01.html
36. Busse I, Deffner B, Schulzrinne H. Dynamic QoS control of multimedia applications based on RTP. *Computer Communications* 1996; **19**(1):49–58.
39. Goyal P, Vin H, Shen C, Shenoy P. A reliable adaptive network protocol for video transport. *Technical Report UTEXAS CS//CSTR*-95-18, The University of Texas at Austin, Department of Computer Sciences, July 1995.
40. Keshav S. REAL: a network simulator. *Technical Report* 88/472, University of California, Berkeley, 1988.
41. Mundur P, Sood A, Simon R. *Network Delay Jitter and Client Buffer Requirements in Distributed Video-on-Demand Systems*. Department of Computer Science, George Mason University, Fairfax, VA 22030.
42. Rejaie R, Handley M, Estrin D. Architectural considerations for playback of quality adaptive video over the Internet. *Technical Report* 98-686, USC-CS, November 1998.
43. Smith HM, Mutka MW, Rover DT. A feedback based rate control algorithm for multicast transmitted video conferencing. *Journal of High Speed Networks* 1999; **7**(3–4):259–279.

AUTHORS' BIOGRAPHIES

Christos Bouras obtained his Diploma and PhD from the Computer Science and Engineering Department of Patras University (Greece). He is currently an Assistant Professor in the above department. He is also a scientific advisor of Research Unit 6 and director of Networking Technologies Sector, in Computer Technology Institute (CTI), Patras, Greece. His research interests include Networks and Protocols, Educational Technology, Telematics and New Services. He has extended professional experience in Design and Analysis of Networks, Protocols, Telematics and New Services. He has published 130 papers in various well-known refereed conferences and journals. He is a co-author of five books in Greek. He has been a PC member and referee in various international journals and conferences. He has participated in R&D projects such as RACE, ESPRIT, TELEMATICS, EDUCATIONAL MULTI-MEDIA, ISPO, EMPLOYMENT, ADAPT, STRIDE, EUROFORM, IST, GROWTH and others. He is also a member of team of experts in the Greek Research and Technology Network (GRNET), ACM, IEEE, EDEN, AACE and New York Academy of Sciences.



Apostolos Gkamas obtained his Diploma and Master Degree from the Computer Engineering and Informatics Department of Patras University (Greece). He is currently an R&D Computer Engineer at the Research Unit 6 of the Computer Technology Institute, Patras, Greece. He is also attending the third year of studies to obtain his PhD degree at the Computer Science and Engineering Department of Patras University (Greece). His research interests include Computer Networks, Telematics, Distributed Systems, Multimedia and Hypermedia. More particular he is engaged in transmission of multimedia data over networks and multicast congestion control. He has published six papers in Journals and 19 papers in well-known refereed conferences. He has participated in R&D projects such as OSYDD, RTS-GUNET, ODL-UP, ODL-OTE, TODAY'S STORIES, ATMNet, 'Technical Consultant to Pedagogical Institute,' ELECTRA and Teleteaching Service for Greek PTT and is currently involved in the project IPv6—GRNET, ASP-NG (ASP-New Game—IST-2001-35354) and 6NET (LARGE-SCALE INTERNATIONAL IPv6 TESTBED—IST-2001-32603).