# Performance of Adaptive Multimedia Transmission: The case of Unicast Technique

Ch. Bouras[1] and A. Gkamas[2]
[1]Research Academic Computer Technology Institute, Greece - Computer Engineering and Informatics Dept., Univ. of Patras, Greece, e-mail: bouras@cti.gr
[2]Research Academic Computer Technology Institute, Greece - Computer Engineering and Informatics Dept., Univ. of Patras, Greece, e-mail: gkamas@cti.gr

## Abstract
Unicast mechanisms for the adaptive transmission of multimedia data can be used for the transmission of multimedia data over heterogeneous networks, like the Internet with the use of one unicast stream for each receiver. Those mechanisms have the capability to adapt the transmission of the multimedia data to network changes. In this paper, we describe a unicast mechanism for adaptive transmission of multimedia data, which is based on real time protocols. The proposed mechanism can be used for unicast 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. Moreover, the proposed mechanism uses a “friendly” to the network users congestion control policy to control the transmission of the multimedia data. We evaluate the adaptive multicast transmission mechanism and we compare it with a number of similar mechanisms available to the literature (LBA, RAP, and TFRCP).

## Keywords

Real time applications, Multimedia communications, Unicast congestion control, Quality of Service issues, TCP friendliness, Communication protocols.

## 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 (Cen et al, 1998). In order to add adaptation characteristics to real time applications, we can use techniques both to the network and application layers. In this paper, we concentrate on the unicast transmission of multimedia over the Internet. The architecture of the proposed mechanism is based on RTP / RTCP (Real time Transmission Protocol / Real time Control Transmission Protocol) (Shculzrinne et al, 1996) for the transmission of the multimedia data. In addition, RTCP offers capabilities for monitoring the transmission quality of multimedia data. The most prominent feature of the proposed mechanism is the network monitoring module which uses a combination of parameters in order to determine the network conditions. Moreover, all the required modules for the implementation of the adaptive streaming mechanism are located on the server side only. We compare the proposed mechanism with other unicast mechanisms available to the literature and we come to the conclusion that the proposed mechanism has satisfactory performance comparing with the other mechanisms. The proposed mechanism has been presented in detail in the following papers: “Bouras and Gkamas, 2000” and “Bouras and Gkamas, 2003”.

The rest of this paper is organised as follows: Section 2 presents the general architecture of a unicast mechanism for the adaptive transmission of multimedia data. Following, section 3 presents the proposed unicast mechanism for the adaptive transmission of multimedia data. In this section we concrete in the feedback analysis and transmission rate estimation modules of the proposed mechanism. Detailed comparison of the proposed mechanism with other unicast mechanisms is presented in section 4. Finally, section 5 concludes the paper and discusses some of our future work.

## 2. Architecture of a unicast mechanism for the adaptive transmission of multimedia data.

The architecture of a unicast mechanism for the adaptive transmission of multimedia data is based on the client – server model. Figure 1 displays the architecture of such mechanism.

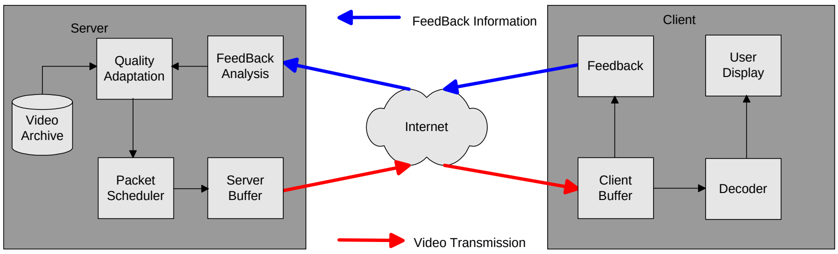



**Figure 1: Architecture of a unicast mechanism for the adaptive transmission of multimedia data**

The server of the unicast mechanism consists of the following modules: ***Video archive:*** Video archive consists of a set of hard disks in which the multimedia 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. ***Quality adaptation:*** It is responsible for the adaptation of the video transmission quality, in order to match with the current network conditions. ***Packet scheduler / Server buffer:*** This module is responsible for the encapsulation of multimedia information in the packets (usually RTP packets). In addition, this module is responsible for the transmission of the RTP packets in the network.

The client of the adaptive streaming multimedia consists of the following modules: ***Client buffer:*** The client application, it stores the incoming data to the buffer before starting present data to the user. ***Feedback:*** This module is responsible for monitoring the transmission quality of the data and informs the server. ***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. The proposed mechanism for the unicast transmission of multimedia data

The proposed mechanism is based on the general architecture of a unicast mechanism for the transmission of multimedia data which presented above. In this section we focus in the feedback analysis module which is responsible to analyse 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. Figure 2 displays the components of the feedback analysis module. The feedback analysis module extracts the packet loss rate and the delay jitter of the RTCP receiver report sent by the client and passes them through the appropriate filters (packet loss rate filter and delay jitter filter respectively).

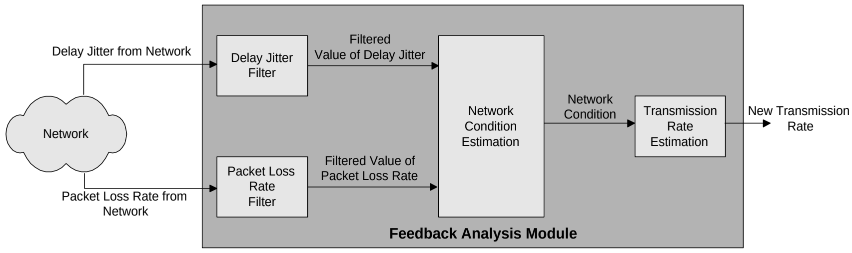



**Figure 2: Feedback Analysis Module**

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

$$LR_{new} = a * LR_{old} + (1-a) * LR_{net} \quad (1)$$

Where: $LR_{new}$: The new filtered value of packet loss rate. $LR_{old}$: The previous filtered value of packet loss rate. When the transmission of the data starts $LR_{old} = 0$. $LR_{net}$: The value of the packet loss rate from the RTCP receiver report that the client sent. $a$: This parameter specifies how aggressive the feedback analysis module will be to the value of the packet loss rate, which receives from the RTCP receiver report. For the parameter $a$ stands $0 \le a \le 1$. The value of the delay jitter passes the following filter:

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

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 $J_{old} = 0$. $J_{net}$: The value of the delay jitter from the RTCP receiver report that the client sent. $\beta$: This parameter specifies how aggressive the feedback analysis module will be to the value of the delay jitter, which it receives from the RTCP receiver report. For the parameter $\beta$ stands $0 \le \beta \le 1$. 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 characterise the network conditions.

The network condition estimation component characterises 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. **Condition unload:** When the network is in unload condition either packet losses does 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 $LR_c$

(congestion packet loss rate) and $LR_u$ (unload packet loss rate), which control the changes among the network conditions based on the following algorithm:

$$\begin{aligned} &if\,(LR_{new} \geq LR_c) \rightarrow congestion \\ &if\,(LR_u < LR_{new} < LR_c) \rightarrow load \\ &if\,(LR_{new} \leq LR_u) \rightarrow unload \end{aligned} \tag{3}$$

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:

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

Where $\gamma$ is a parameter, which specifies how aggressive the network condition estimation component will be to the increase of delay jitter. In other words $\gamma$ specifies quantitatively the expression “abrupt increase of delay jitter”.

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. More particularly the transmission estimation module uses following algorithm:

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

Where: $R_{new}$: The new value of the transmission rate. $R_{old}$: The old value of the transmission rate. $R_{increase}$: The factor with which the server increases the transmission rate in the case of available bandwidth. $R_{decrease}$: The factor with which the server decreases the transmission rate in the case of network congestion. For this factor stands: $0 < R_{decrease} < 1$. 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. A major result of our experiments is that the choice of the above parameters is a trade off. When we changed one parameter in order to improve the behaviour of the proposed mechanism to one point, the behaviour of the proposed mechanism worsen to other points. The choice of the above parameters depends on the network and especially on the dominant traffic in the network.

## 4. Comparing proposed mechanism with other unicast mechanisms

In this section, we compare the performance of proposed mechanism with other unicast mechanisms with adaptation capabilities found to the literature. We compare these mechanisms based on the following parameters: TCP friendliness, stability, and convergence time to stable state. The above parameters set outline well the behavior of a unicast adaptation scheme for the transmission of multimedia data. We compare the proposed mechanism with the following unicast mechanisms: (1) LBA (Sisalem, 1998): LBA stands for "Loss Based Adjustment Algorithm" and it is a mechanism for the unicast transmission of multimedia data which is based on the use of RTP / RTCP protocols. The use of LBA decreases packet losses to the network and increases the utilization of the available bandwidth. (2) RAP (Rejaie et al, 1999): RAP stands for "Rate Adaptation Protocol" and it is a mechanism for the unicast transmission of multimedia data which is based on the use of AIMD (Additive Increase Multiplicative Decrease) algorithm. RAP mechanism focuses on TCP friendliness issues. (3) TFRCP (Pandhye et al, 1999): TFRCP stands for "TCP - Friendly Rate Control Protocol" and it is a mechanism for the unicast transmission of multimedia data which collects information regarding packet losses and RTT time and based on that information it uses the model presented on (Pandhye et al, 1998) in order to estimate TCP friendly transmission rates.

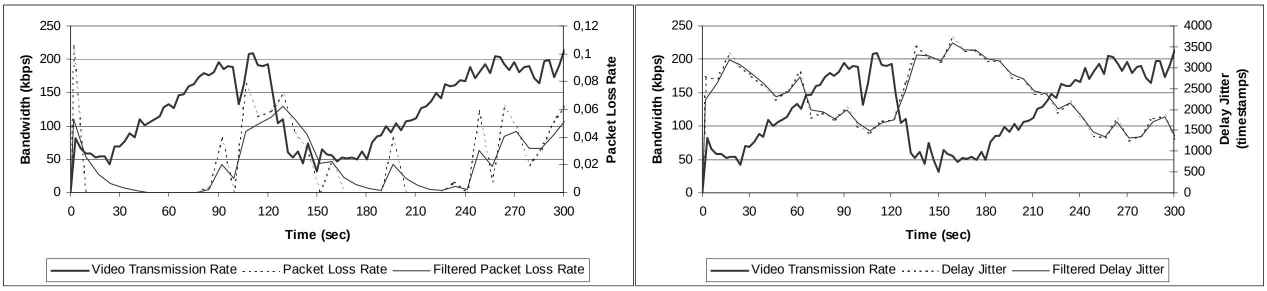



**Figure 3: Proposed mechanism transmission rate packet loss rate and delay jitter during experiment**

Figure 3 displays the proposed mechanism transmission rate, the packet loss rate and the delay jitter during the transmission of TCP traffic and traffic produced by the proposed mechanism over a bottleneck link. During this experiment, we transmit at the same time multimedia data with the use of the proposed mechanism and TCP traffic through the bottleneck link. In order to realise our experiments to a stable network environment, we used the following 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 streaming application. The VC had capacity 300 kbps and at the same time we transmitted video with the use of adaptive streaming application and TCP traffic in order to evaluate the behaviour of the adaptive streaming application. We use the following scenario: Initially, we transmit only video to the bottleneck link with the use of the proposed mechanism. Two minutes after the beginning of multimedia transmission, we transmit TCP traffic through the bottleneck link together with the adaptive multimedia. The transmission of adaptive multimedia continues for one minute until the video files ends, after the end of TCP traffic. The proposed mechanism has a transmission rate of 50kbps, when the multimedia transmission starts. We use the following values for the parameters that control the behaviour of proposed mechanism: $\alpha = 0.5$, $\beta = 0.8$, $\gamma = 2$, $LR_c = 0.05$, $LR_u = 0.02$, $R_{increase} = 20.000$ (bps), $R_{decrease} = 0.50$. As someone can see in Figure 3, when the transmission of TCP traffic starts and network congestion occurs, proposed mechanism releases bandwidth in order to be used by the TCP traffic. Consecutively, the proposed mechanism keeps steady its transmission rate until the transmission of TCP stops. Then the proposed mechanism gradually reserves again all the available bandwidth. From the Figure 3, it is obvious that both packet loss rate and delay jitter indicates network congestion. In addition, the proposed mechanism has fluctuations in the transmission rate and it can not keep its transmission rate stable. Moreover, the proposed mechanism has moderate performance regarding TCP friendliness, but on the other hand, the proposed mechanism does not starve the TCP traffic and the TCP traffic has good performance with the existence of traffic produced by the proposed mechanism.

Figure 4 shows the bandwidth allocation during the transmission of TCP traffic and traffic produced by LBA mechanism. The scenario of the experiment includes sharing of network link among a TCP connection and two LBA connections. As Figure 4 shows, LBA mechanism has not friendly behavior against TCP traffic and the traffic produced by LBA mechanism starves the TCP traffic. In addition, LBA mechanism has a significant stable operation and the convergence time to stable operation is small but LBA mechanism has significant delay to consume the available bandwidth. Comparing the LBA behaviour with the proposed mechanism behaviour, we can draw the following conclusions: LBA mechanism has better performance regarding mechanism stability and the proposed mechanisms has better behaviour regarding the convergence time to stable operation and TCP friendliness.

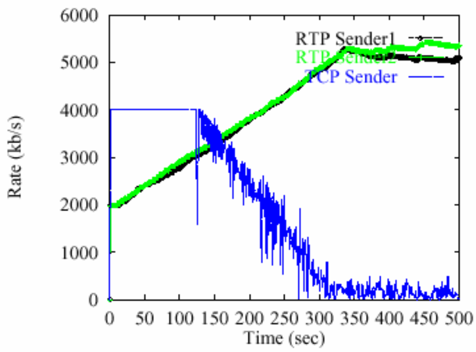
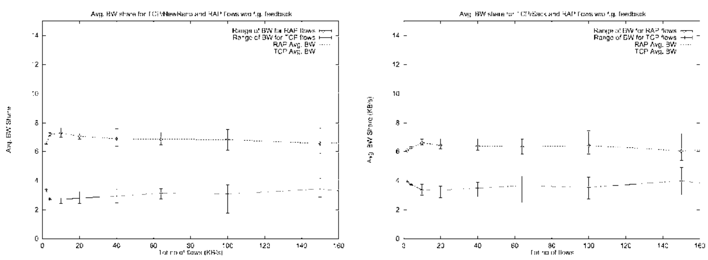



**Figure 4: LBA performance against TCP traffic**



(a) RAP coexisting with NewReno

(b) RAP coexisting with Sack

**Figure 5: RAP performance against TCP traffic**

Figure 5 shows the bandwidth allocation during the transmission of TCP traffic and traffic produced by RAP mechanism. The scenario of the experiment includes sharing of network link among $n$ TCP connection and $n$ RAP connections. As Figure 5 shows, RAP mechanism has a significant stable operation and very small convergence time to stable operation. RAP mechanism has moderate performance regarding TCP friendliness and RAP traffic consumes more bandwidth than TCP traffic but TCP traffic does not starves. In addition, RAP mechanism has stable transmission rate during the entire experiment. Comparing the RAP behaviour with the proposed mechanism behaviour we can draw the following conclusions: RAP mechanism has better performance regarding mechanism stability and convergence time to stable operation comparing with the proposed mechanisms. Both mechanisms has moderate performance regarding TCP friendliness and RAP has more stable operation and proposed mechanism has less aggressive behavior and allow TCP traffic to receive more bandwidth.

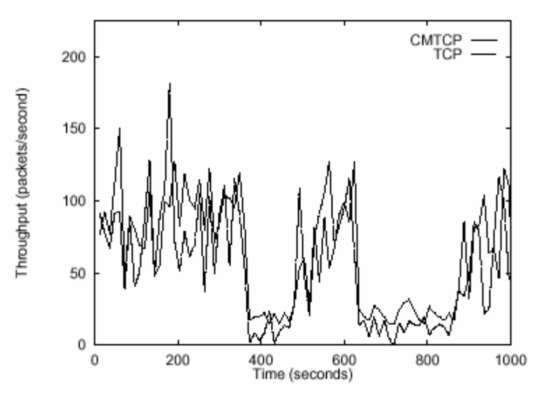



**Figure 6: TFRCP performance against TCP traffic (in the figure TFRCP is called - Continues Media TCP)**

| Parameter / Mechanism | Proposed mechanism | LBA | RAP | TFRCP |
|---|---|---|---|---|
| **TCP friendliness** | Moderate | Bad | Moderate | Good |
| **Stable transmission rate** | No | Yes | Yes | No |
| **Convergence time** | Relative fast | Moderate | Relative fast | Moderate |
| **Stable operation** | Moderate | Yes | Yes | No |

**Table 1: Comparison of the proposed mechanism with the other unicast mechanism for the transmission of multimedia data**

Figure 6 shows the bandwidth allocation during the transmission of TCP traffic and traffic produced by TFRCP mechanism. The scenario of the experiment includes the transmission of a TCP connection and a TFRCP connection over the Internet. As Figure 6 shows RAP mechanism has friendly behavior against TCP traffic and both TCP traffic and traffic produced by TFRCP mechanism have almost the same bandwidth share during the entire experiment. In addition, TFRCP mechanism does not have stable operation and it does not keep its transmission stable (TCP traffic has also the same behavior). We have to mention that the fact that the experiment took place over the Internet where the network conditions are not controlled may lead to the above behavior. Comparing the TFRCP mechanism behaviour with the proposed mechanism behaviour we can draw the following conclusions: The proposed mechanism has better behavior regarding mechanism stability and convergence time to stable operation comparing with the TFRCP mechanism. Regarding TCP friendliness, TFRCP mechanism has a significant friendly behavior against TCP traffic and has better performance comparing with the proposed mechanism. TFRCP mechanism has that significant friendly behavior against TCP traffic due to the fact that TFRCP mechanism is using a very accurate

TCP model (presented at Pandhye et al, 1998) for the estimation of TCP friendly transmission rate.

Table 1 summarizes the comparison of the proposed mechanism against the others unicast mechanism for the transmission of multimedia data. As this table shows the proposed mechanism has moderate performance regarding TCP friendliness, but on the other hand, the proposed mechanism does not starve the TCP traffic and the TCP traffic has good performance with the existence of traffic produced by the proposed mechanism. General speaking the proposed mechanism has satisfactory performance comparing with the other mechanisms. The main drawback of the proposed mechanism is the fact that the proposed mechanism has fluctuations in the transmission rate and it can not keep its transmission rate stable.

## 5. Conclusion - Future work

In this paper, we are concentrating to the design of a mechanism for monitoring the network condition and estimating the appropriate rate for the transmission of the multimedia data in order to treat with fairness the receivers. In addition, we compare the proposed mechanism with other unicast mechanism. Main conclusion of this comparison is that the proposed mechanism has good performance and its main drawbacks are the fluctuations in the transmission rate and its unstable transmission rate.

Our future work includes experiments in the Internet and simulations in a network simulation environment, in order to evaluate the behaviour of the adaptive streaming application to the combination of traffics types in a changeable network environment. Moreover, we plan to enhance the proposed mechanism in order to use multicast and also a TCP model during the transmission rate estimation in order to increase TCP friendliness and try to eliminate transmission rate fluctuations.

## 6. References

Bouras, C. and Gkamas, A. (2003), "Multimedia Transmission with Adaptive QoS based on Real Time Protocols", *International Journal of Communications Systems*, Wiley InterScience, 16, pp. 225 - 248.

Bouras, C. and Gkamas, A. (2000), "Streaming multimedia data with adaptive QoS characteristics", *Protocols for Multimedia Systems*, Cracow, Poland, October 22 - 25 2000, pp. 129 - 139.

Cen, S. Pu, C. and Walpole, J. (1998), "Flow and Congestion Control for Internet Media Streaming Applications", *In Proceedings of Multimedia Computing and Networking*.

Shculzrinne, H. Casner, S. Frederick, R. and Jacobson, V. (1996) "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, IETF.

Pandhye, J. Kurose, J. Towsley, D. and R. Koodli, (1999) "A model based TCP - friendly rate control protocol", *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Basking Ridge, NJ.

Pandhye, J. Firoiu, V. Towsley, D. and Kurose, J. (1998) "Modeling TCP throughput: a simple model and its empirical validation", ACMSIGCOMM.

Rejaie, R. Handley, M. and Estrin, D. (1999) "RAP An end - to - end rate - based congestion control mechanism for realtime streams in the Internet", *Proc. IEEE Infocom*.

Sisalem, D. (1998) "Fairness of adaptive multimedia applications", *IEEE International Conference on Communications. Conference Record. Affiliated with SUPERCOMM'98 IEEE.*