Comparing performance of SRAMT-LE vs other layered encoding schemes regarding TCP friendliness

Christos Bouras, Apostolos Gkamas

Research Academic Computer Technology Institute, 61 Riga Feraiou Str, GR-26221 Patras, Greece
Computer Engineering and Informatics Dept., Univ. of Patras, GR-26500 Patras, Greece
e-mail: {bouras, gkamas}@cti.gr

Abstract. In this paper we describe a hybrid sender and receiver-based adaptation scheme for multicast transmission of multimedia data using layered encoding, which we call SRAMT-LE (Sender-Receiver based Adaptation scheme for Multicast Transmission using Layered Encoding). The most prominent features of SRAMT-LE are its distributed (to sender and receivers) transmission rate estimation algorithm and its innovative RTT (Round Trip Time) estimation algorithm based on one-way delay measurements. SRAMT-LE is using both a TCP model and an AIMD (Additive Increase Multiplicative Decrease) algorithm in order to estimate a TCP friendly bandwidth share. We evaluate SRAMT-LE and compare it with a number of similar layered encoding schemes available to the literature (PLM, RLC, MLDA). Main conclusion of this evaluation was that SRAMT-LE has friendly behavior against the dominant traffic types of today's Internet and has a relative good behavior comparing with the other layered encoding schemes available to the literature.

1. Introduction

The 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 network environment that Internet provides to real time applications as well as the lack of sufficient QoS (Quality of Service) guarantees, many times forces applications to embody adaptation schemes in order to work efficiently. In addition, any application that transmits data over the Internet should have a friendly behavior towards the other flows that coexist in today's Internet and especially towards the TCP flows that comprise the majority of flows. We define as TCP friendly flow, a flow that consumes no more bandwidth than a TCP connection, which is traversing the same path with that flow ([12]).

The methods proposed for the multicast transmission of multimedia data over the Internet can be generally divided in three main categories, depending on the number of multicast streams used: (1) The sender uses a single multicast stream for all receivers ([1], [3], [16]). (2) Simulcast: The sender transmits versions of the same video, encoded in varying degrees of quality. This results to the creation of a small number of multicast streams with different transmission rates ([8], [6], [4]). (3) The sender uses layered encoded video, which is video that can be reconstructed from a number of discrete data layers, the basic layer and more additional layers, and transmits each layer into different multicast stream ([10], [9], [14], [15]).

In this paper, we briefly present an adaptation scheme for multicast transmission of multimedia data over best effort networks, like the Internet, which provides the most satisfaction to the group of receivers, with the current network conditions. We call this adaptation scheme SRAMT-LE (Sender-Receiver based Adaptation scheme for Multicast Transmission using Layered Encoding) and it is a hybrid sender and receiver-based adaptation scheme. SRAMT-LE is trying to transmit TCP friendly multicast flows with the use of layered encoding video. SRAMT-LE creates n layers (the basic layer and n-1 additional layers) and transmits each layer in different multicast streams, each one within certain bandwidth limits. The basic layer provides the basic video quality and each additional layer improves the video quality. A receiver in order to be able to decode the video layers and present the video information must receive the layer k and also the layers 1-(k-1) and then we say that the receiver is in layer subscription level k. More information regarding SMART-LE and a detail evaluation of SMART-LE can be found in [5]. In this paper we give also a detail comparison of the SMART-LE with other layered encoding schemes available to the literature. Main target of this comparison is to compare the SMART-LE performance of the performance of other layered encoding schemes available to the literature against the following criteria: TCP friendliness, Stability, Scalability and Convergence time to stable state. The above parameters set outline well the behavior of a layered encoding congestion control scheme.

2. Description of SRAMT-LE

With the use of SRAMT-LE, the sender transmits multimedia data to a group of m receivers with the use of multicast. Sender is using the layered encoding approach, and transmits the video information in n different layers (the basic layer and $n-1$ additional layers). The receivers join the appropriate number of layers, which better suit their requirements.

2.1. Sender Operation

The sender generates n different layer managers. Each layer manager is responsible for the transmission of a video layer. Each receiver manager corresponds to a unique receiver. In addition, the synchronization server is responsible for the management, synchronization and intercommunication between layer managers and receiver managers. We have added an application specific part (APP) to the RTCP receiver reports, which the receivers sent to the RTP/RTCP session of the basic layer, in order to include the receivers' estimation about the TCP friendly bandwidth share $r^i_{r\_tcp}$ in the path between the receiver and the sender, the packet loss rate estimation l_i in all layers, which this receiver is listening and the receiver layer subscription level (the maximum layer up to which the receiver is listening) k. Receiver managers store the last value of $r^i_{r\_tcp}$, l_i and k from the receiver, which represent, and these information is used for the adjustment of layers transmission rates.

When a receiver manager receives a RTCP receiver report from the receiver i (which represents) is using the packet loss rate l_i to estimate the transmission rate r^i_{AIMD} of the receiver i with the use of an AIMD (Additive Increase Multiplicative Decrease) algorithm (which has been presented in [2]). In addition, the receiver manager is using the analytical model of TCP presented in [12] in order to estimate a TCP friendly bandwidth share $r^i_{l\_tcp}$ in the path between the receiver and the sender: If the receiver experiences packet losses, a TCP friendly bandwidth share $r^i_{l\_tcp}$ (in bytes/sec) is estimated with the use of the equation (1) (where t^{r-i}_{RTT} is the sender estimation for RTT between that receiver and the sender), and l_i is the packet loss rate that the receiver i reports):

$$r^i_{l\_tcp} = \frac{P}{t^{r-i}_{RTT}\sqrt{\frac{2l_i}{3}} + 4t^{r-i}_{RTT}\min(1, 3\sqrt{\frac{3l_i}{8}})l_i(1+32l_i^2)} \quad \textbf{(1)}$$

If the receiver does not experience packet losses, in order to estimate a TCP friendly bandwidth share $r^i_{l\_tcp}$, the $r^i_{l\_tcp}$ must not be increased more than a packet / RTT. For this reason receiver manager calculates the new value of $r^i_{l\_tcp}$ by adding $(T_{rr}\ /t^{r-i}_{RTT})$ packets (where T_{rr} is the time space between the current and the last receiver report of receiver i) to the previous value of $r^i_{l\_tcp}$ (the $r^i_{l\_tcp}$ is expressed in bytes/sec):

$$r^i_{l\_tcp} = r^i_{l\_tcp} + \frac{T_{rr}}{(t^{r-i}_{RTT})^2}P \quad \textbf{(2)}$$

Then the receiver manager selects as receiver's i preferred transmission rate r^i the minimum of the $r^i_{r\_tcp}$, r^i_{AIMD}, $r^i_{l\_tcp}$:

$$r^i = \min(r^i_{r\_tcp}, r^i_{AIMD}, r^i_{l\_tcp}) \quad \textbf{(3)}$$

Each time one receiver manager receives a receiver report in the basic layer session form the receiver, which represents, informs synchronization manager in order to adjust the layers' transmission rates. The adjustment of layers transmission rates has as target to produce TCP friendly cumulative transmission rate for any layer subscription level k. For this reason the synchronization manager polls the r^i values of the receivers that are listening only to basic layer (layer 1) and sets as transmission rate of layer 1 $r_{layer-1}$ the minimum value of r^i of the receivers that are listening only to basic layer. Then polls the r^i values of the receivers that are listening up layer 2 and sets as transmission rate of layer 2 $r_{layer-2}$ the minimum values of r^i minus the $r_{layer-1}$. This procedure repeats for all the layers:

$r_{layer-1} = \min(r^i)$ for all receiver i listening up to layer 1 (basic layer) **(4)**

$r_{layer-2} = \min(r^i) - r_{layer-1}$ for all receiver listening up to layer 2

…

$r_{layer-n} = \min(r^i) - r_{layer-n-1}$ for all receiver i listening up to layer n

In addition, the sender includes to all the RTP packets, which transmits, the transmission rate of all the layers. This information can be used from the receivers in order to change their subscription level and accommodate better their requirements.

2.2 Receiver Operation

Each receiver measures the following parameters of the path, which connects it with the sender: (1) Packet loss rate (l_i): The receiver calculates the packet loss rate during the reception of sender layers based on RTP packets sequence numbers. (2) RTT estimations (t_{RTT}^{e-i}): The receiver makes an estimation for the RTT between it and the sender based on one way delay measurements with the use of RTP packets timestamps. The receiver emulates the behavior of a TCP agent with the use of the analytical model of TCP presented in [12] and estimates a TCP friendly bandwidth share $r^i_{r\_tcp}$ every RTT time. If the receiver experiences packet losses is using the following equation in order to estimate a TCP friendly bandwidth share (in bytes/sec):

$$r^i_{r\_tcp} = \frac{P}{t_{RTT}^{e-i}\sqrt{\frac{2l_i}{3}} + 4t_{RTT}^{e-i}\min(1,3\sqrt{\frac{3l_i}{8}})l_i(1+32l_i^2)} \quad (5)$$

If the receiver does not experience packet losses, in order to estimate a TCP friendly bandwidth share $r^i_{r\_tcp}$, the $r^i_{r\_tcp}$ must not be increased more than a packet / RTT. For this reason receiver calculates the value of $r^i_{r\_tcp}$ with the following equation (in bytes/sec):

$$r^i_{r\_tcp} = r^i_{r\_tcp} + \frac{1}{t_{RTT}^{e-i}}P \quad (6)$$

Each time the receiver sends a receiver report to the sender, using the RTP/RTCP session of the basic layer, includes the average value of $r^i_{r\_tcp}$ since last receiver report. In addition the receiver has the capability to add or remove layers based on the information that gathers itself and the information that sender includes in to RTP packets. The receivers' layer subscription changes are synchronized at the end of a specific time period T_{epoch}, which we call epoch. The receiver change their layer subscription (add or remove layers) using the following procedure: At the end of each epoch, each receiver compares the value of the $r^i_{r\_tcp}$, with the cumulative transmission rates of the sender layers and change its layer subscription level up to layer k in order to satisfy the following constraint:

$$r^i_{r\_tcp} <= \sum_{j=1}^{k} r_{layer-j} \quad (7)$$

We declare as unsuccessful layer change the situation when a receiver joins (or leaves) a layer and after a sort time period (T_{change}) drop (or add) again this layer. During our performance evaluation, we observe that the unsuccessful layer changes by the receivers cause instability to the operation of SRAMT-LE and must be avoided. In order to avoid unsuccessful layer changes by the receivers, when a receiver makes an unsuccessful layer change we avert the receiver to make the layer change, which was unsuccessful, for the next $2^k * T_{change}$ time (where k the number of continuant unsuccessful layer changes since the last successful layer change). Due to fact that T_{change} affects linearly the value $2^k * T_{change}$ and the k affects the value of $2^k * T_{change}$ exponentially, we set T_{change} to 5 seconds but other values of T_{change} can also be used.

2.3 SMART-LE details

In this paragraph we present some details regarding the operation of SMART-LE (more detailed information can be found in [5]):

– *Packet Loss Rate Estimation:* In order to prevent a single spurious packet loss having an excessive effect on the packet loss estimation, receivers smooth the values of packet loss rate using the following filter, which computes the weighted average of the m most recent loss rate values $l_{i,j}^{m}$.
– *RTT Estimations:* When a receiver i receives a RTP packet from a sender layer, uses an algorithm based on one way measurements in order to estimate the Round Trip Time (RTT) between the sender and the receiver.
– *Extensions to RTP/RTCP:* RTP provides an extension mechanism to allow individual implementations that require additional information to be carried in the RTP data packet header. SRAMT-LE uses the extension mechanism of RTP in order to add to the additional fields in to RTP header and new application specific part (APP) to the RTCP reports.
– *Synchronization of stream changes:* Similar research has shown ([10]) that, if the receivers synchronize their layer changes, the synchronization problems can be minimized. For this reason the receivers' layer changes are synchronized in the end of each epoch.
– *Scalability issues:* In order to ensure that, when the group of the participants increases, the sender will collect feedback information representing all the receivers, we use the partial suppression method proposed in [11] to control the transmission of the RTCP reports.

3. Comparing SRAMT-LE with other layered encoding schemes

In this section we compare the performance of SRAMT-LE mechanism with other mechanism founded to the literature regarding the following parameters: TCP friendliness, Stability, Scalability and Convergence time to stable state. The above parameters set outline well the behavior of a layered encoding congestion control scheme. We compare the SMART-LE with the following layered encoding schemes:

– *PLM* ([9]): PLM stands for “Packet pair receiver-driven Layered Multicast” and is based on a cumulative layered scheme and on the use of packet pair to infer the bandwidth available at the bottleneck to decide which are the appropriate layers to join. PLM assumes that the routers are multicast capable but does not make any assumption on the multicast routing protocol used. PLM is receiver driven, so all the burden of the congestion control mechanism is at the receivers side. The only assumption we make on the sender is the ability to send data via cumulative layers and to emit for each layer packets in pairs (two packets are sent back-to-back). PLM is highly scalable due to the receiver-driven cumulative layered scheme. PLM does not require either any signaling or feedback.
– *MLDA* ([14]): MLDA stands for “Multicast enhanced Loss-Delay based Adaptation algorithm”. MLDA is a hybrid sender and receiver-based adaptation scheme that combines on the one hand various well known concepts for multicast congestion control such as receiver-based rate calculation, layered transmission and dynamic into a unified congestion control architecture. Scalability in MLDA is based on partial suppression method.
– *RLC* ([15]): RLC stands for “Receiver-driven, Layered Congestion control algorithm”. RLC is designed to support one-to-many communication to potentially large sets of receivers with different bandwidth requirements. RLC uses a hierarchical, layered scheme for data transmission, where receivers can join to one or more multicast groups to receive data at a rate approximately matching their bandwidth to the source - this translates into different quality levels in the case of multimedia streams, or in faster transfer times for reliable data communication. Scalability in RLC comes from full decentralization of functionality: each receiver takes congestion control decisions autonomously.

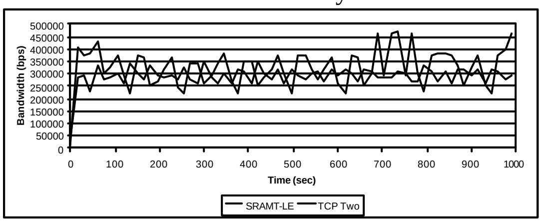

Fig. 1. Bandwidth distribution on C1-C3 bottleneck link

Figure 1 shows the bandwidth distribution to bottleneck link shared by SRAMT-LE and a TCP connection with the same RTT time. As this figure indicate that SRAMT-LE is in general fair towards to TCP connections and treats the heterogeneous group of the receivers with fairness. SRAMT-LE behaves as is expected, and shares the available bandwidth with the TCP connection with the same RTT delay. The behavior of SRAMT-LE (“seeking” for available bandwidth and reaction to congestion) leads some times to get more bandwidth share than TCP and some times to get less bandwidth share than TCP, but in long term both the SRAMT-

LE and the TCP flows get the approximately the same bandwidth share of the bottleneck links.

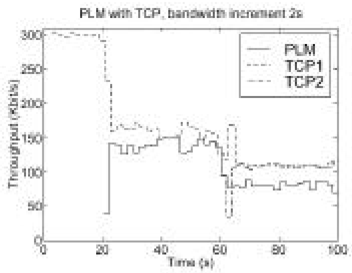

Fig. 2. PLM performance against TCP traffic

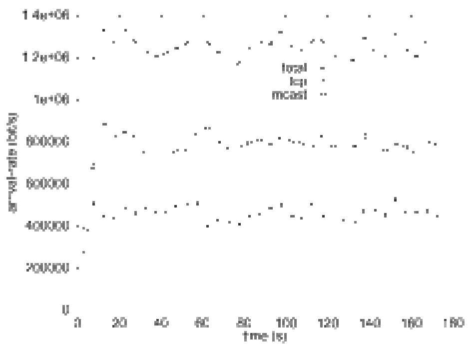

Fig. 3. RLC performance against TCP traffic

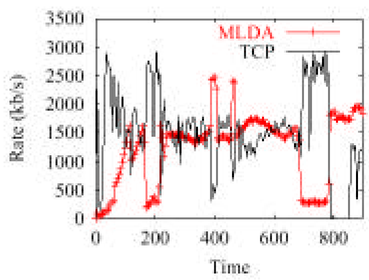

Fig. 4. MLDA performance against TCP traffic

Figure 2 shows how the PLM shares a bottleneck link initially with one TCP connections and later on with two TCP connections. The simulation scenario was the following: Initially the first TCP connections transmits data and at the 20th second starts the transmission of the PLM session and finally at the 60th second starts the transmission of the second TCP connection over the bottleneck link. As figure 2 shows, the PLM session adapts all most perfectly to the available bandwidth in presence of TCP flows. Comparing the PLM behavior with the SMART-LE behavior we can draw the following conclusions: PLM has more stable transmission rate and change its transmission rate in steps comparing with SMART-LE which can not keep its transmission rate stable and changed it continues during the entire experiment (figure 1). In long term, we can say that in the case of PLM, TCP traffic gets more bandwidth that the PLM traffic but in the case of SMART-LE, TCP and SMART-LE traffics are share almost them equally the available bandwidth. In order to summarize, both PLM and SMART-LE have good behavior against the TCP traffic with PLM offering a more stable transmission rate and SMART-LE offering more fair bandwidth sharing. In addition, the PLM has a fast convergence time to the stable state after the transmission of the TCP traffic to the bottleneck link. The main disadvantage of PLM is the fact that assumes that the routers of network testbed support some kind of a fair queuing mechanism that allocates each flow a fair bandwidth share. Only under this assumption, is it possible to use PLM for congestion control. The fact that the Internet router does not support fair queuing mechanisms at the moment (and it is not expected to support fair queuing mechanisms in large scale to the near future) has as result the difficult large scale deployment of PLM to the Internet.

Figure 3 shows how the RLC shares a bottleneck link with TCP traffic. The simulation scenario includes the transmission of 8 RLC sessions together with 8 TCP connections over a bottleneck link. As figure 3 shows, RLC is slightly more aggressive than TCP, but this was expected as RLC considers closely spaced losses as a single event, whereas TCP does not. On the other hand, TCP and RLC do not starve each other when competing. Comparing the RLC behavior with the SMART-LE behavior we can draw the following conclusions: Both RLC and SMART-LE have some fluctuation on their transmission rates but they keep relative stable their transmission rates. In addition, it is obvious that SRAMT-LE has more friendly behavior against TCP traffic than RLC has. In addition, both RLC and SMART-LE have similar convergence times to the stable state. In order to summarize, SMART-LE has better behavior against the TCP traffic comparing with RLC and this is because the TCP analytical model used by SMART-LE is more accurate than the TCP analytical model used by the RLC. On the other hand, the RLC has a much more simple implementation comparing with SMART-LE.

Figure 4 shows how the MLDA shares a bottleneck link with TCP traffic. Figure 4 shows the bandwidth share between MLDA and TCP traffic in the bottleneck link. As figure 4 shows, the MLDA has friendly behavior against TCP traffic most of the simulation time but in some cases either the TCP traffic starves MLDA traffic or MLDA traffic starves TCP traffic (most of the starve cases). Comparing the MLDA behavior with the SMART-LE behavior we can draw the following conclusions: The SMART-LE behavior is friendlier that MLDA behavior against TCP traffic mainly due to the fact the SMART-LE traffic does not starve TCP traffic as MLDA traffic does in some cases. In addition, MLDA has long convergence times to the stable state comparing with SMART-LE. Moreover, both MLDA and SMART-LE do not keep their transmission rates stable but they have fluctuation on their transmission rates. In order to summarize, both MLDA and SMART-LE have similar behavior but the MLDA has the drawback of big convergence time to the stable state and starving of TCP traffic in some cases.

Table 1 summarizes the comparison of SMART-LE against the others layered encoding schemes. As this table shows, SMART-LE has good performance against TCP traffic and in general terms has good performance comparing with the other layered encoding schemes. The main drawback of the SMART-LE mechanism is the fact that SMART-LE has fluctuation on its transmission rate and it is not keep its transmission rate stable. This has as result the TCP connections also to have fluctuation on their transmission rates as reaction to the continues changing network conditions due to the above mentions SMART-LE behavior.

Table 1. Comparison of SMART-LE with the other layered encoding schemes

Parameter / Mechanism	**SMART-LE**	**PLM**	**RLC**	**MLDA**
TCP friendliness	Very Good	Good	modest	Good
Stable transmission rate	No	Yes	Yes	No
Convergence time	Relative fast	Very fast	Relative fast	Modest
Stable operation	Yes	Yes	Yes	No
Limitations	No	fair queuing mechanism in routers	No	No
Scalability	Well - partial suppression method	Well - not require feedback for the client	Well - not require feedback for the client	Well partial suppression method

4. Conclusion - Future Work

In this paper, we present the behavior investigation of the SRAMT-LE, a mechanism for multicast transmission of adaptive multimedia data in a heterogeneous group of receivers with the use of layered encoding. SRAMT-LE is using a hybrid sender and receiver-based adaptation scheme and uses both a TCP model and an AIMD algorithm to estimate a TCP friendly bandwidth share. We investigate the behavior of SRAMT-LE through a number of simulations. We compare also the behavior of SRAMT-LE with other layered encoding schemes available to the literature. Main conclusion of the evaluation was that SRAMT-LE has friendly behavior against the dominant traffic types (TCP traffic) of today's Internet and good behavior during congestion condition. In addition SRAMT-LE provides good performance comparing with other layered encoding schemes available to the literature.

Our future work includes the investigation of the fluctuations in SRAMT-LE transmission rate in order the SMART-LE to transmit more smooth transmission rates. In addition, we plan to investigate to dynamically adding more layers instead of the static number of layers that SRAMT-LE supports now. Moreover we plan to implement a prototype of SRAMT-LE and evaluate its operation over the real Internet and compare the results of the Internet evaluation with the simulation results, which are presented in this paper.

References

[1]. J.-C. Bolot, T. Turletti, I. Wakeman, "Scalable Feedback Control For Multicast Video Distribution In The Internet", In Proceedings of SIGCOMM 1994, London, England, August 1994, pp. 139-146.
[2]. Ch. Bouras, A. Gkamas, "Streaming Multimedia Data With Adaptive Qos Characteristics", Protocols for Multimedia Systems 2000, Cracow, Poland, October 22-25, 2000, pp 129-139.
[3]. C. Bouras, A. Gkamas, "A Mechanism For Multicast Multimedia Data With Adaptive Qos Characteristics", 6th International Conference on Protocols for Multimedia Systems-PROMS 2001, Enschede, The Netherlands, October 17-19 2001, pp. 74-88.
[4]. Ch. Bouras, A. Gkamas, An. Karaliotas, K. Stamos, "Architecture And Performance Evaluation For Redundant Multicast Transmission Supporting Adaptive Qos", 2001 International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2001), Split, Dubrovnik (Croatia) Ancona, Bari (Italy), October 09-12, 2001.
[5]. C. Bouras, A. Gkamas, "SRAMT-LE: A Hybrid Sender And Receiver-Based Adaptation Scheme For TCP" Communication Networks and Distributed Systems Modelling and Simulation Conference (CNDS-03 Part of the 2003 Western MultiConference), Marriott Orlando Airport, Orlando, Florida, January 19 - 23 2003, pp. 71-76
[6]. S. Y. Cheung, M. Ammar, X. Li, "On the Use of Destination Set Grouping to Improve Fariness in Multicast Video Distribution", INFOCOM 96, San Fransisco, March 1996.
[7]. S. Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, vol. 1,4: pp. 397-413, 1993.
[8]. T. Jiang, E. W. Zegura, M. Ammar, "Inter-Receiver Fair Multicast Communication Over The Internet", In Proceedings of the 9th International Workshopon Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), June 1999, pp. 103-114.
[9]. Legout and E. W. Biersack. "PLM: Fast convergence for cumulative layered multicast transmission schemes". In Proceedings of ACM SIGMETRICS'2000, Santa Clara, CA, USA, June 2000
[10]. S. McCanne, V. Jacobson, "Receiver-Driven Layered Multicast", 1996 ACM Sigcomm Conference, August 1996, pp. 117-130.
[11]. J. Nonnenmacher, Ernst W. Biersack, "Optimal Multicast Feedback", in Proceedings of the Conference on Computer Communications (IEEE Infocom), San Francisco, USA, Mar. 1998.
[12]. J. Pandhye, J. Kurose, D. Towsley, R. Koodli, "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, June 1999.
[13]. H. Shculzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, IETF, January 1996.
[14]. D. Sisalem, A. Wolisz, "MLDA: A TCP-Friendly Congestion Control Framework For Heterogeneous Multicast Environments", in Eighth International Workshop on Quality of Service (IWQoS 2000), Pittsburgh, PA, June 2000.
[15]. L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer", in Proceedings of the Conference on Computer Communications (IEEE Infocom), San Francisco, USA, Mar. 1998.
[16]. J. Widmer, M. Handley, "Extending Equation-Based Congestion Control To Multicast Applications", in ACM SIGCOMM, August 2001.