# SRAMT-LE: A Hybrid Sender And Receiver-Based Adaptation Scheme For TCP Friendly Multicast Transmission Using Layered Encoding

**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**

**Keywords:** Multicast congestion control, Simulation of multimedia applications, Quality of service issues, TCP friendliness, Simulation of communication protocols.

## 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. In addition, SRAMT-LE adjusts the transmission rates of the layers in order to service better the group of the receivers. 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. With the use of SRAMT-LE, we ensure that sender will transmit TCP friendly traffic and receivers with different capabilities (in terms of available bandwidth) are able to receive the multimedia information. We evaluate SRAMT-LE through a number of simulations in order to examine its behaviour to a heterogeneous group of receivers and its behaviour against TCP connections. Main conclusion of the simulations was that SRAMT-LE has friendly behaviour against the dominant traffic types of today's Internet and treats a heterogeneous group of receivers with fairness.

## 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 nature of the Internet makes the multicast transmission of real time multimedia data a challenge. Different receivers of the same multicast stream may have different processing capabilities, different loss tolerance and different bandwidth available in the paths leading to them.

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 behaviour 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 ([14]).

During the multicast transmission over the Internet, several aspects need to be considered: (1) Transmission rate adaptation: The sender must adapt the transmission rate based on the current network conditions. (2) TCP friendliness: During the multicast transmission over the Internet, the multicasts flows must be TCP-friendly. (3) Scalability: The performance of the adaptation scheme must not be deteriorated with increasing numbers of receivers. (4) Heterogeneity: The adaptation scheme needs to take into account the heterogeneity of the Internet and must aim at satisfying the requirements of a large part of the receivers if not all possible receivers.

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:

- The sender uses a single multicast stream for all receivers ([1], [3], [19]). This results to the most effective use of the network resources, but on the other hand the fairness problem among the receivers arises especially when the receivers have very different capabilities.
- 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 ([9], [5], [4]). Each receiver joins the stream that carries the video quality, in terms of transmission rate, that it is capable of receiving.
- 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 ([11], [17]). The basic layer provides the basic quality and the quality improves with each additional layer. The receivers subscribe to one or more multicast streams depending on the available bandwidth into the network path to the source.

The subject of transmission TCP friendly flows over networks has engaged researchers all over the world ([14], [17], [19]). Various adaptation schemes deploy an analytical model of TCP ([14]) in order to estimate a TCP friendly bandwidth share. With the use of this model, the average bandwidth share of a TCP ($r_{tcp}$) connection is determined (in bytes/sec) with the following equation:

$$r_{tcp} = \frac{P}{t_{RTT}\sqrt{\frac{2Dl}{3}} + t_{out}\min(1,3\sqrt{\frac{3Dl}{8}})l(1+32l^2)} \quad (1)$$

Where $P$ is packet size in bytes, $l$ is the packet loss rate, $t_{out}$ is the TCP retransmission timeout, $t_{RTT}$ is the RTT (Round Trip Time) of the TCP connection and $D$ the number of acknowledged TCP packets by each acknowledgment packet. SRAMT-LE is using the above described analytical model of TCP, in order to estimate TCP friendly bandwidth shares. For the following of this paper we assume that $D=1$ (each acknowledgment packet acknowledges one TCP packet) and $t_{out} = 4t_{RTT}$ (the TCP retransmission timeout is set to be four time the RTT).

In this paper, we propose 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. The most prominent features of SRAMT-LE, comparing with other adaptation schemes based on layer encoding, which have already been presented in the literature, are: (1) the dynamic adjustment of layer transmission rates (2) the innovative RTT (Round Trip Time) estimation algorithm based on one-way delay measurements (3) the combination of various methods (TCP model, AIMD, etc) for the estimation receivers' preferred transmission rates.

# SRAMT-LE ARCHITECTURE

## General

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 Sender transmits each layer into a different RTP/RTCP multicast session. The transmission rate within each layer is adapting within its limits (each layer has an upper and lower limit in its transmission rate) according to the capabilities of the receivers listening up to it. The receivers join the appropriate number of layers which better suit their requirements (available bandwidth between the sender and the receiver, etc) and if during the transmission of multimedia data the network conditions to the path between them and the sender change, the receivers have the capability to receive more or less video layers in order to accomplish better their requirements. The communication between the sender and the receivers is based on RTP/RTCP sessions and the sender is using the RTP protocol to transmit the video layers and the participants (the sender and the receivers) use the RTCP protocol in order to exchange control messages. In the following paragraphs, we give details about the different aspects of SRAMT-LE.

## Sender Operation

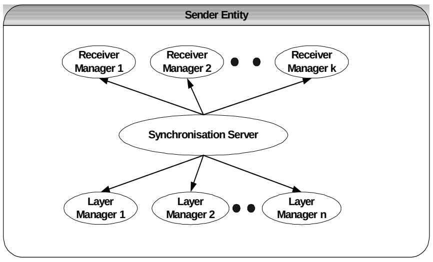


**Figure 1. The architecture and the data flow of the Sender.**

Figure 1 shows the organisation and the architecture of the SRAMT-LE sender entity. The sender generates $n$ different layer managers. Each layer manager is responsible for the transmission of a video layer. The sender creates a new receiver manager every time receives a RTCP report from a new receiver. Each receiver manager corresponds to a unique receiver. It processes the RTCP reports generated by the receiver and can be considered as a representative of the receiver at the side of the sender. In addition, the synchronisation server is responsible for the management, synchronization and intercommunication between layer managers and receiver managers. If a receiver manager does not receive RTCP reports from the receiver, which represents for long time, stops its operation and releases its resources.

With the use of RTCP adaptive feedback mechanism the receivers send their feedback to the sender in the form of RTCP receivers reports. 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) $k$ (more information in section "Receiver Operation" and section "Extensions to RTP/RTCP"). 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 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 (more information in section "RTT Estimations")), and $l_i$ is the packet loss rate that the receiver $i$ reports (more information in section "Packet Loss Rate Estimation")):

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

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):

$$r^i_{l\_tcp} = r^i_{l\_tcp} + \frac{T_{rr}}{(t^{r-i}_{RTT})^2}P \quad (3)$$

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

$$r^i = \min(r^i_{r\_tcp}, r^i_{AIMD}, r^i_{l\_tcp}) \quad (4)$$

Each time one receiver manager receives a receiver report in the basic layer session form the receiver, which represents, informs synchronisation 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 synchronisation 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:

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

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

(5)

…

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

With the use of the above procedure, we ensure that sender will transmit TCP friendly traffic and in addition, due to the fact that the transmission rate of the basic layer is set to the minimum value of receiver preferred transmission rates, SRAMT-LE ensures that all the receiver will be able to receive multimedia information whereas their available bandwidth is low comparative with the available bandwidth of other receivers (We have to mention that during the transmission of multimedia data, there is a lower limit in the required available bandwidth, under which the receiver is not able to receive enough multimedia information and has to stop receiving the multimedia data).

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.

## Receiver Operation

Each receiver measures the characteristics of the path, which connects it with the sender and informs the sender with the use of receiver reports. Each receiver measures the following parameters of the path, which connects it with the sender:

- Packet loss rate ($l_i$): The receiver calculates the packet loss rate during the reception of sender layers based on RTP packets sequence numbers (more information in section Packet Loss Rate Estimation).
- 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 (more information in section RTT Estimations).

The receiver emulates the behavior of a TCP agent with the use of the analytical model of TCP and estimates a TCP friendly bandwidth share $r^i_{r\_tcp}$ every RTT time using equation (1). If the receiver experiences packet losses is using the following equation in order to estimate a TCP friendly bandwidth share (in bytes/sec):

$$r^i_{r\_tcp} = \frac{P}{t_{RTT}^{e-i}\sqrt{\frac{2l_i}{3}} + 4t_{RTT}^{e-i}\min(1,3\sqrt{\frac{3l_i}{8}})l_i(1+32l_i^2)} \quad (6)$$

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):

$$r^i_{r\_tcp} = r^i_{r\_tcp} + \frac{1}{t_{RTT}^{e-i}}P \quad (7)$$

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 (more information in section Extensions to RTP/RTCP). The receivers' layer subscription changes are synchronized at the end of a specific time period $T_{\text{epoch}}$, 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:

$$r^i_{r\_tcp} <= \sum_{j=1}^{k} r_{layer-j} \quad (8)$$

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.

# SRAMT-LE PARAMETERS ESTIMATION

## Packet Loss Rate Estimation

Each receiver measures the packet loss rate based on RTP packets sequence numbers in each layer (the sender transmits each layer in different RTP/RTCP session). 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 $l^m_{i,l}$ (the following filter has been presented in [19] and has been evaluate and gives a good estimation of packet loss rate):

$$l_{i,l} = \frac{\sum_{j=0}^{m-1} w_j l_{i,l}^{m-j}}{\sum_{i=0}^{m-i} w_i} \text{ for receiver } i \text{ and layer } l \quad (9)$$

Where $l_{i,l}$ is the smooth value of packet loss rate for layer $l$. The weights $w_i$ are chosen so that very recent packet loss rates receive the same high weights, while the weights gradually decrease to 0 for older packet loss rate values. In our simulations we use m=8 and the following values for the weights $w_i$: {1,1,1,1,0.8,0.6,0.4,0.2}. The packet loss rate, for all the layers ($1..k$) that the receiver receives, is calculated with the following equation:

$$l_i = \frac{\sum_{j=1}^{k} l_{i,j} * r_{layer-j}}{\sum_{j=1}^{k} r_{layer-j}} \quad (10)$$

## RTT Estimations

When a receiver $i$ receives a RTP packet from a sender layer, uses the following algorithm in order to estimate the Round Trip Time (RTT) between the sender and the receiver. If we assume that the sender and the receiver have synchronized clocks, receiver can use the timestamp of the RTP packet ($T_{timestamp}$) and the local time that receives that packet ($T_{receiver}$) in order to estimate the one way delay form sender to receiver ($T_{oneway}$):

$$T_{oneway} = T_{receiver} - T_{timestamp} \quad (11)$$

If the path between the sender and the receiver was symmetric and the path had the same delay into both directions, the RTT between the sender and the receiver would be twice the $T_{oneway}$:

$$t_{RTT} = 2T_{oneway} \quad (12)$$

Until now, we have made two assumptions: (1) the sender and the receiver have synchronized clocks (2) the path between the sender and the receiver is symmetric. The above assumptions are not true for the Internet and as results in order to get accurate RTT estimations ($t_{RTT}^{e-l}$), receivers have to take the above assumptions into account. For this reason, we use a parameter $a$ and we can write the equation (12) as:

$$t_{RTT}^{e-l} = (1+a)T_{oneway} \quad (13)$$

The parameter $a$ is used in order to smooth the estimation of the RTT due to the potential unsynchronized clocks between the receiver and the sender and due to the potential asymmetry of the path between the sender and the receiver. In order to avoid solely phenomenon to affect the RTT estimations, receivers pass the $t_{RTT}^{e-l}$ values through a filter similar to the filter, which they use for filtering the values of packet loss rate (more information in section Packet Loss Rate Estimation).

In order to estimate the value of parameter $a$, receivers need an effective estimation of RTT, which can be acquired, with the use of RTCP reports: The RTCP receiver report contains the $t_{LSR}$ (the timestamp of most recent RTCP sender report from the sender) and $t_{DLSR}$ (The delay between receiving the last sender report from sender and sending this receiver report) values. As result the sender can made an effective RTT measurement for the path between it and a receiver by using the following equation (where $A$ is the time which the sender receives the receiver report from that receiver):

$$t_{RTT}^{r-i} = A - t_{LSR} - t_{DLSR} \quad (14)$$

The sender estimates an effective RTT measurement for a receiver $i$ every time receives a receiver report from that receiver and includes this effective RTT measurement (with the id of the receiver) in the next RTP packet of the basic layer.

A receiver after receives an effective RTT measurement from the sender, estimates an appropriate value for the parameter $a$ using the following equation:

$$a = \frac{t_{RTT}^{r-i}}{T_{oneway}} - 1 \quad (15)$$

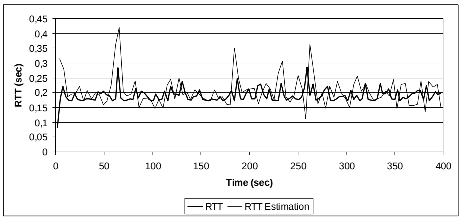


**Figure 2. RTT Estimations**

Figure 2 shows the real values of RTT and the values, which obtained with the above RTT estimations algorithm during the transmission of multimedia data with the use of SRAMT-LE over a 1Mbit link with background traffic produced by an on/off traffic generator using an exponential distribution with transmission rate of 0.5 Mbps during on times. This figure shows that in most of the cases the above algorithm give a good approximation of the real RTT values.

The values of $t_{RTT}^{e-l}$ give an estimation of RTT based on measurement on each layer $l$. The receiver is using for TCP friendly transmission rate estimation the average value of $t_{RTT}^{e-l}$ for all the layers ($1..k$) that receives:

$$t_{RTT}^{e-i} = \frac{\sum_{l=1}^{k} t_{RTT}^{e-l}}{k} \quad (16)$$

# EXTENSIONS TO RTP/RTCP

As we have already mentioned, the operation of SRAMT-LE is based on the transmission with the use of 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 following fields in to RTP header:

- $T_{\text{epoch}}$: The specific time period called epoch, in which the receivers have the capability to change their layer subscription level.
- $t_{RTT}^{r-i}$ and receiver id: With this field the sender informs the receiver $i$ about the effective RTT measurement between this receiver and the sender.
- Current transmission rates of sender layers $r_{layer-j}, j = 1,..,n$.
- End of epoch flag: This flag is used in order the receivers to be informed about the end of an epoch and synchronize their layer subscription level change actions.

In addition, RTCP gives the capability to the participants to include in the RTCP reports an application specific part (APP) intended for experimental use. The receivers add to their receiver report an application specific part, which contains the average value of their estimations for TCP friendly bandwidth share $r^i_{r\_tcp}$ and the packet loss rate estimation $l_i$ in all layers, which this receiver is listening, since last receiver report. Moreover the receivers add to their receiver report the current layer subscription (the maximum layer up to which the receives listening) $k$.

Due to the fact that all the participants listening at least to the RTP/RTCP session of basic layer, the above described extensions to RTP/RTCP are used only to the basic layer RTP/RTCP session. The RTP/RTCP protocols with the incorporation of the above described extensions can support in whole the operation of SRAMT-LE.

# SYNCHRONIZATION OF STREAM CHANGES

During the multicast transmission of data, a multicast stream transverses a network node as long as at least one receiver behind that node is listening to that stream. As result, if a receiver stop listening to a layer, the transmission of the layer will stop only if that receiver was the only one receiver listening to that layer behind that node. In addition, if two receivers join different layers at the same time, the receiver which joins the layer with the lower (cumulative) transmission rate might observe losses that were not caused by his action but by the action of receiver join the layer with the higher (cumulative) transmission rate.

Similar research has shown ([11]) that, if the receivers synchronize their layer changes, the above problems can be minimized. For this reason the receivers' layer changes are synchronized in the end of each epoch. The sender marks the next RTP packet in the basic layer (all the receivers are listening at least to the basic layer) after the end of an epoch with a special flag, which indicates the end of the epoch. However due to the network heterogeneity and packet losses, some receivers may not receive the special marked packet, or receive it in different time points. For this reason the sender includes the epoch duration $T_{\text{epoch}}$ in all the RTP packets that transmits. Receivers can change streams either when receive a special marked packet or after ($T_{\text{epoch}} + T_{oneway}$) time after the end of the previous epoch (where $T_{oneway}$ is the one way (sender to receiver) delay estimation of that receiver). During our simulation we set the $T_{\text{epoch}}$ to be 5 seconds in order to allow receivers to quickly find the subscription level which fulfils better their requirements. The small value of $T_{\text{epoch}}$ does not cause problems due to the tracing and suspending of unsuccessful stream changes mechanism that SRAMT-LE supports (more information in section Receiver Operation).

# SCALABILITY ISSUES

The RTCP adaptive transmission mechanism defines as 5 sec the minimum value for RTCP report retransmission timeout. The RTCP adaptive transmission mechanism has as result the interval between the RTCP reports (each participant sends) to increase when the group of the participant increases.

In order to ensure that, when the group of the participants increases, the sender will collect feedback information representing all the receivers, we do the following modification to the RTCP adaptive transmission mechanism: When the RTCP adaptive transmission mechanism suggests a big retransmission interval more that $T_{suspent}$ (which means that the number of participants has increase too), the receivers is using the partial suppression method proposed in [13] to control the transmission of the RTCP reports. According to that partial suppression method, the receivers are using a truncated exponentially distributed retransmission timer in the interval $[0, T_{rand}]$ with density of:

$$T_{wait} = \frac{1}{\exp^{\lambda} - 1} * \frac{\lambda}{T_{rand}} \exp^{\frac{\lambda}{T_{rand}} z} \text{ if } 0 \le z \le T_{rand} \quad (17)$$

$$T_{wait} = 0 \text{ otherwise}$$

Each receiver schedules the RTCP retransmission timeout to be $T_{wait}$. If the receiver listens in the multicast RTP/RTCP session of the basic layer, a receiver report from an other receiver with TCP friendly bandwidth share $r^i_{r\_tcp}$ similar to its estimation of TCP friendly bandwidth share (we consider that two TCP friendly bandwidth shares are similar when they differ up to 2%), this receiver suspend the transmission of its receiver report. As [13] shows analytically, with the appropriate selection of the equation (17) parameters ($\lambda, T_{rand}$), for 10.000 receivers less than 10 feedback messages are generated for each event the receivers are reporting on. During our simulations we set $T_{suspent}$ to 10 sec in order to ensure that the sender will always have feedback information, which represents all the receivers. With the above described mechanism, when the number of the receiver is small the sender collects information from all the receivers. When the number of the receivers is big the sender collects information from a part of receivers, which represents all the receivers.

# MECHANISM EVALUATION

In this section, we present a number of simulations that we made in order to analyze the behavior of SRAMT-LE, during the multicast transmission of multimedia data with the use of layered encoding approach. We implemented SRAMT-LE and run simulations in the LBNL network simulator ns-2 ([12]).

## Heterogeneous Multicast Environments - TCP friendliness

In this simulation, we investigate the performance of SRAMT-LE in a heterogeneous multicast environment and its TCP friendliness. We choose to investigate the TCP friendliness of SRAMT-LE in a multicast distribution tree without any shared links among the receivers. With this approach, we investigate the TCP-friendliness of SRAMT-LE without having to consider the effects of interaction between different receivers, traversing multiple routers and different round trip delays among the receivers.

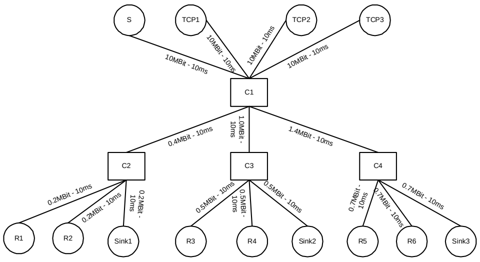


**Figure 3. Topology with no share links**

Figure 3 shows the topology of this simulation. The bandwidth of each link is given to the simulation topology and varies from 0.2 Mbps to 10.0 Mbps. All the links in the simulation topology are full duplex, have delay 10 ms and they use the RED (Random Early Drop) ([7]) policy to their queues. With the use of RED, we assure that all the flows receive the same loss ration and we avoid the synchronization among the flows. In this topology we have one sender (S), which transmits multimedia data with the use of SRAMT-LE to a group of 6 receivers (R1-R6) with different capabilities. The sender transmits three layers with the following limits: layer one (basic layer): 50Kbps-200Kbps, layer two: 50Kbps-400Kbps and layer three: 50Kbps-400Kbps. With this configuration the maximum cumulative transmission rate up to layer one is 200, up to layer two is 600Kbps and up to layer three is 1000Kbps. In addition we have 3 TCP sources TCP1, TCP2, TCP3, which transmit data to Sink1, Sink2, Sink3. We model the TCP sources as "4.3BSD Tahoe TCP" ([18]) sources, which always have data to send during the simulation. In the simulation topology we have three bottleneck links (C1-C2, C1-C3 and C1-C4) and each router (C2, C3, C4) is shared between sender layers and a TCP connection with the same RTT time as the sender / receiver pair.

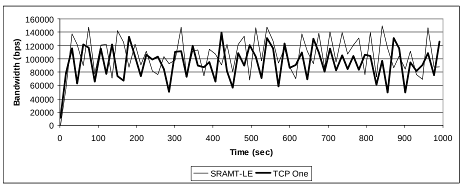


**Figure 4. Bandwidth distribution on C1-C2 bottleneck link**

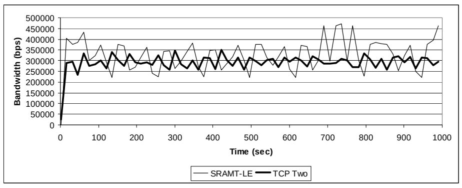


**Figure 5. Bandwidth distribution on C1-C3 bottleneck link**

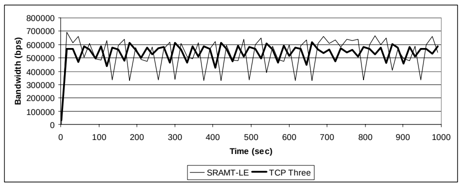


**Figure 6. Bandwidth distribution on C1-C4 bottleneck link**

We execute this simulation for 1000 seconds and the sender starts transmitting all the layers with transmission rate 150Kbps. With the above describe topology we expect that Receivers R1 and R2 will join only the basic layer (layer subscription level 1), receivers R3 and R4 will join up to layer two (layer subscription level 2), and receivers R5 and R6 will join up to layer three (layer subscription level 3).

Figure 4, Figure 5 and Figure 6 shows the bandwidth distribution to bottleneck links C1-C2, C1-C3 and C1-C4 respectively. Receivers behave as we except and each receiver has the expected layer subscription level. These figures indicate that SRAMT-LE is in general fair towards to TCP connections and treats the heterogeneous group of the receivers with fairness. In all bottleneck links 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.

## Multicast Environments With Share Links

In this simulation, we investigate the performance of SRAMT-LE in a heterogeneous multicast environment with a multicast distribution tree that is shared among the receivers. With this approach, we investigate the behavior of SRAMT-LE, when the actions of one receiver affect other receivers.

Figure 7 shows the topology of this simulation. The bandwidth of each link is given to the simulation topology and varies form 0.2 Mbps to 10.0 Mbps. All the links in the simulation topology are full duplex, have delay, which varies form 10 ms to 70 ms, and they use again the RED policy to their queues. In this topology we have one sender (S), which transmits multimedia data with the use of SRAMT-LE to a group of 5 receivers (R1-R5) with different capabilities. The sender transmits again three layers with the following limits: layer one (basic layer): 50Kbps-200Kbps, layer two: 50Kbps-400Kbps and layer three: 50Kbps-400Kbps. With this configuration the maximum cumulative transmission rate up to layer one is 200, up to layer two is 600Kbps and up to layer three is 1000Kbps. In addition each of the links C1-C2, C2-C3 and C3-C4 is shared between the sender layers and an uncorrelated background traffic, which consumes maximally the 50% of the link capacity. In order to produce the uncorrelated background traffic, we use a traffic generator with active and idle periods. During the active periods the transmission rate of the traffic generator follows a Pareto distribution with a scale factor of 1.1 and a mean of 20 packets. Active transfer phases are then followed by idle periods drawn by a Pareto distribution with a scale factor of 1.8 and a mean 0.5 seconds. As [15] suggests the above traffic generator models background web traffic.

We execute this simulation for 1000 seconds and the sender starts transmitting all the layers with transmission rate 150Kbps. In order to avoid synchronization, the receivers join randomly the layer one during the first 3 seconds of the simulation. With the above describe topology we expect that

Receivers R5 and R3 will join only the basic layer (layer subscription level 1), receivers R2 and R4 will join up to layer two (layer subscription level 2), and receivers R1 will join up to layer three (layer subscription level 3).

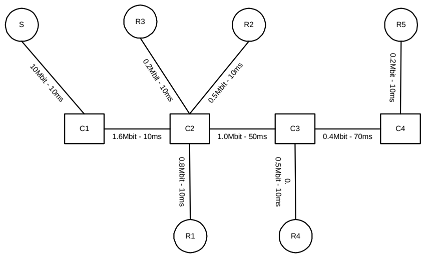


**Figure 7. Topology with share links**

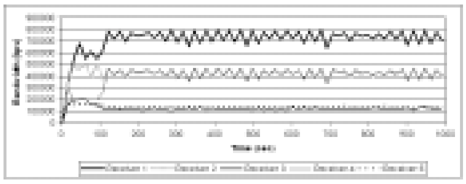

**Figure 8. Bandwidth shares of Receiver R1 to Receiver R5**

Figure 8 shows the bandwidth share of the receivers R1 to R5 during this simulation. As this figure suggests after some seconds each receiver has the layer subscription level, which we expect and receives also a bandwidth share close to the bandwidth share, which we expect. The only exception is the transmission rate of layer subscription lever one which is close to 120Kbps and not close to 200Kbps, which is the expected transmission rate, based on the topology of Figure 7. The explanation for that is the following: Because the multicast stream of the basic layer is the layer with the biggest delay (due to the hops S, C1, C2, C3, C4, C5) and biggest loss rate (due to the fact that the layer one pass three congested links C1-C2, C2-C3 and C4-C5) the estimations of TCP friendly bandwidth share of receiver R5 is low and the sender keeps the transmission rate of basic layer low in order to service also the receiver R5. The receivers after some unsuccessful stream changes (during the first 100 seconds) have join the layers which fulfills better their capabilities and stay at that stream until the end of the simulation (due to the tracing of unsuccessful layer changes that SRAMT-LE offers). In addition, due to the synchronization of layer changes the undesirable problems are minimal and in general the receivers actions does affect the bandwidth shares of the other receivers.

## 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. Main conclusion of the simulation 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 treats with fairness a heterogeneous group of receivers.

Our future work includes the investigation of 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] 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.

[6] C. Diot, "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.

[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] S. Floyd, K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", *IEEE/ACM Transactions on Networking*, 1998.

[9] 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.

[10] T. Kim, M. H. Ammar, "A Comparison Of Layering And Stream Replication Video Multicast Schemes", *Proc. NOSSDAV'01*, Port Jefferson, NY, June 25-26, 2001.

[11] S. McCanne, V. Jacobson, "Receiver-Driven Layered Multicast", 1996 *ACM Sigcomm Conference*, August 1996, pp. 117-130.

[12] S. McCanne, S. Floyd, "The UCB/LBNL Network Simulator", software online. http://www.isi.edu/nsnam/ns/.

[13] J. Nonnenmacher, Ernst W. Biersack, "Optimal Multicast Feedback", in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, San Francisco, USA, Mar. 1998.

[14] 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.

[15] K. Park, G. Kim, M. Crovella, "On the Relationship Between File Sizes, Transport Protocols, and Self-Similar Network Traffic", *In Proceedings of the International Conference on Network Protocols,* pp 171-180, October, 1996.

[16] H. Shculzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, IETF, January 1996.

[17] 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.

[18] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery Algorithms", RFC 2001, January 1997.

[19] J. Widmer, M. Handley, "Extending Equation-Based Congestion Control To Multicast Applications", *in ACM SIGCOMM*, August 2001.