Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Networks 47 (2005) 551–575

Computer Networks

www.elsevier.com/locate/comnet

SRAMT: a hybrid sender and receiver-based adaptation scheme for TCP friendly multicast transmission

Ch. Bouras *, A. Gkamas [1]

Research Academic Computer Technology Institute, Riga Feraiou 61, GR-26221 Patras, Greece
Computer Engineering and Informatics Department, University of Patras, GR-26500 Patras, Achaia, Greece

Received 21 November 2003; received in revised form 13 September 2004; accepted 14 September 2004
Available online 11 November 2004

Responsible Editor: G. Palazzo

Abstract

In this paper, we describe a hybrid sender and receiver-based adaptation scheme for multicast transmission of multimedia data, which we call SRAMT (Sender–Receiver based Adaptation scheme for Multicast Transmission). The most prominent features of SRAMT 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. With the use of SRAMT, 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 through a number of simulations and compare it with other schemes available to the literature. Main target of the simulations was the examination of SRAMT behavior to a heterogeneous group of receivers and its behavior against TCP connections.

Keywords: Multicast congestion control; Quality of Service issues; TCP friendliness; Communication protocols; Multimedia applications

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 nature of the Internet makes the multicast transmission of real time multimedia data a challenge. Different

* Corresponding author. Address: CTI, P.O. BOX 1122, GR-26110, Patras, Greece. Tel.: +30 2610 960375/996951; fax: +30 2610 960358/969016.

E-mail addresses: bouras@cti.gr (Ch. Bouras), gkamas@cti.gr (A. Gkamas).

[1] Tel.: +30 2610 960355; fax: +30 2610 960358.

1389-1286/$ - see front matter
doi:10.1016/j.comnet.2004.09.007

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 behavior towards the other flows that co-exist 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 [18].

The implementation of adaptation mechanisms in the applications is often criticized. The main arguments that rise against it, are that the technologies that are used today for the implementation of the core networks provide capabilities to support QoS; as a result the network should offer to the applications QoS guarantees. This is generally true but there is a big problem about it: today's Internet is divided into thousands of different administration domains. The QoS strategies that are implemented on each one are certainly different (for example QoS based on DiffServ Concept [16], QoS based on IntServ Concept [5], QoS based on IPv6 infrastructure [7]), and in many cases no QoS strategy is implemented at all. So the multimedia data flows that have to traverse many of these different domains in order to reach the end user do not have a sufficient QoS support. The proposed mechanism provides an adaptation service which does not require any QoS support from the network, and runs in any IP multicast network. Another idea widely supported among network administrators, is that the cost of exhaustive monitoring of the network as well as the upgrade of the links that constrain the entire network domain (bottlenecks and critical links) cost less than the deployment of QoS schemes (research, testing and personnel training) [8].

Many researchers urge that due to the use of new technologies for the implementation of the networks, which offer QoS guarantees, adaptive real time applications will not be used in the future. We believe that this is not true and adaptive real time applications will be used in the future for the following reasons: (1) Users may not always want to pay the extra cost for a service with specific QoS guarantees, when they have the capability to access a service with good adaptive behavior, (2) some networks may never be able to provide specific QoS guarantees to the users, (3) even if the Internet eventually supports reservation mechanisms or differentiated services, it is more likely to be on per-class than per-flow basis. Thus, flows are still expected to perform congestion control within their own class. (4) With the use of the differential services network model in the future, networks will support services with QoS guarantees together with best effort services and adaptive services.

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 multicast 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 if not all possible receivers.

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 (Sender–Receiver based Adaptation scheme for Multicast Transmission) and it is a hybrid sender and receiver-based adaptation scheme. SRAMT is trying to transmit TCP friendly multicast flows. We propose two variations of SRAMT: (1) SRAMT-Simulcast (SRAMT-S) which is using simulcast approach for the transmission of multicast data and (2) SRAMT-Layered Encoding (SRAMT-LE) which is using layered encoding approach for the transmission of multicast data. SRAMT-S creates n

different multicast streams (in most network conditions a small number of different multicast streams is enough—typically 3 or 4 multicast streams), each one within certain bandwidth limits. All the multicast streams carry the same multimedia information, each one of them having a different quality and as result different transmission rate. SRAMT-LE creates n layers (the basic layer and $n-1$ addition 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 addition 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 number of the streams/layers depends on the number of receivers with different reception capabilities that expected to receive the multimedia information from the sender and the processing capabilities of the computer where the sender runs. The upper and lower limit of each stream/layer depends on the encoding which is used and the encoder capabilities.

The most prominent features of SRAMT, comparing with other adaptation schemes, which have already been presented in the literature, are: (1) the dynamic adjustment of sender transmission rate (both of in SRAMT-S and SRAMT-LE variations), (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.

The rest of this paper is organised as follows: Section 2 presents some related work available to the literature. Section 3 presents the architecture of the of the SRAMT mechanism. Section 4 provides details on how the SRAMT estimates its parameters. Section 5 describes the necessary additions to RTP/RTCP in order to support the operation of SRAMT mechanism. Section 6 provides information about the synchronization issues of stream changes and Section 7 provides information about scalability issues. In Section 8, we present the performance evaluation of the SRAMT mechanism. In Section 9, we compare the performance of SRAMT with other schemes available to the literature. Finally, Section 10 discusses some of our future work and Section 11 concludes the paper.

2. Related work

When someone multicast multimedia data over the Internet has to accommodate receivers with heterogeneous data reception capabilities. To accommodate heterogeneity, the sender application may transmit one multicast stream and determine the transmission rate that satisfy most the receivers, may transmit at multiple multicast streams with different transmission rates and allocate receivers at each stream or may use layered encoding and transmit each layer to a different multicast stream. It is important for adaptive real time applications to have "friendly" behavior to the dominant transport protocols of today's Internet [10].

Single multicast stream approaches has the disadvantage that clients with a low bandwidth link will always get a high-bandwidth stream if most of the other members are connected via a height bandwidth link and the same is true for the other way around. Someone can overcome the above described problem with the use of a multi-stream multicast approach. Single multicast stream approaches have the advantages of easy encoder and decoder implementation and simple protocol operation, due to the fact that during the single multicast stream approach there is not any need for synchronisation of receivers' actions (as the multiple multicast streams and layered encoding approaches require).

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 [1,3,23,24]. 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 [11,6,4]. The different multicast streams carry the same video information but in each one the video is encoded with different bit rates, and even different video formats (MPEG, H263, JPEG). Each receiver joins in the stream that carries the video quality, in terms of transmission rate, that it is capable of receiving. The main disadvantage in this case is that the same multimedia information is replicated over the network but recent research has shown that under some conditions simulcast has better behavior that multicast transmission of layered encoded video [12].
- 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 [13,14,21,23]. 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 of TCP friendly flows over networks has engaged researchers all over the world [18,21,23]. Various adaptation schemes deploy an analytical model of TCP [18] in order to estimate a TCP friendly bandwidth share. With the use of this model, the average bandwidth share (r_{tcp}) of a TCP connection is determined (in bytes/s) with the following equation:

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

where P is packet size in bytes, l is the packet loss rate, t_{out} is the TCP retransmission timeout, t_{RTT} is the Round Trip Time (RTT) of the TCP connection and D the number of acknowledged TCP packets by each acknowledgment packet. SRAMT 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 $t_{\mathrm{out}} = 4t_{\mathrm{RTT}}$ (the TCP retransmission timeout is set to be four time the RTT).

3. SRAMT architecture

3.1. General

With the use of SRAMT, the sender transmits multimedia data to a group of m receivers with the use of multicast. Sender is using the simulcast approach (SRAMT-S) or layered encoding approach (SRAMT-LE), and transmits the video information in n different streams (SRAMT-S) or n different layers (the basic layer and $n-1$ additional layers) (SRAMT-LE). The sender transmits each stream/layer into a different RTP/RTCP [20] multicast session. The transmission rate within each stream/layer is adapting within its limits (each stream/layer has an upper and lower limit in its transmission rate) according to the capabilities of the receivers. The receivers join the appropriate streams/layers which suit better 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 change stream, or to receive more or less video layers in order to accomplish better their requirements.

Based on our experience and SRAMT evaluation we come to the conclusion that when someone what to implement SRAMT to a real network he must use as many streams/layers as the number of the different network connections which are used by the end users. In practice the number of different network connections which are used today are relative small (PSTN, ISDN, ADSL, Cable modem, LAN, ...) and this lead to a relative small number of streams/layers needed by SRAMT. If the SRAMT administrator have more information about the end users can assign more that one network connection technologies (for example PSTN and GPRS that have similar transmission rates) to a streams/layers in order to decrease more the number of streams/layers needed. One other issue that the SRAMT administrator has to take into account is to select appropriate encoding for each stream/layer. For example the SRAMT administrator must select a low bit encoding for a stream, which will be received by receivers with low network connection (for example H.263 for ISDN connections) and the opposite (MPEG-2 for LAN connections). Regarding the limits of each steam/layer on safe solution is to set as max transmission rate for a stream the max transmission rate of the network connection that the intended receivers are use and to set as min transmission rate the max transmission rate of the next lower stream/layer. Moreover the SRAMT administrator must ensure that the SRAMT transmission entity has enough processing power in order to perform all the necessary encodings and its network connection can support the transmission of all the stream/layers at the same time.

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 streams/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 mechanism.

3.2. Sender operation

Fig. 1 shows the organisation and the architecture of the SRAMT sender entity. The sender generates n different stream managers (SRAMT-S) or n different layer managers (SRAMT-LE). Each stream/layer manager is responsible for the transmission of a video stream/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 stream/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 receiver reports. We have added an application specific part (APP) to the RTCP receiver reports, which are sent by the receivers, in order to include the receivers' estimation about the TCP friendly bandwidth share $r^{i}_{r\_\mathrm{tcp}}$ in the path between the receiver and the sender, the packet loss rate estimation l_i, the stream number k

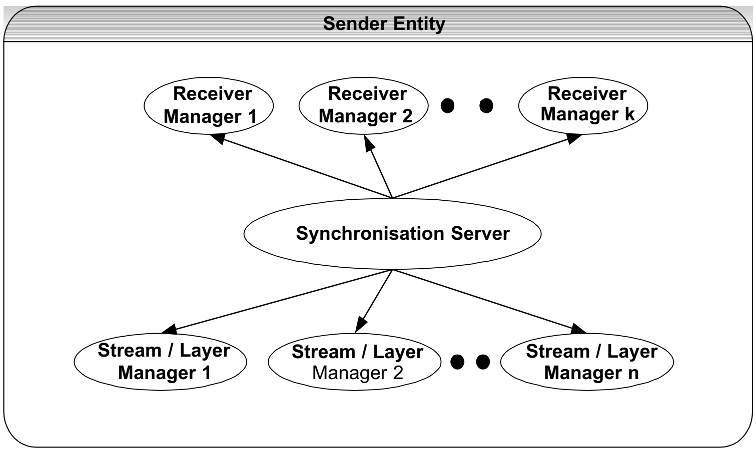

Fig. 1. The architecture and the data flow of the sender.

which the receiver receives (SRAMT-S) or receiver layer subscription level (the maximum layer up to which the receives listening) k (SRAMT-LE). Receiver manager stores the last value of $r^i_{r\_\mathrm{tcp}}$, l_i and k from the receiver, which represents, and this information is used for the adjustment the sender transmission rate.

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 $r^i_{l\_\mathrm{tcp}}$ in the path between the receiver and the sender: if the receiver experiences packet losses, a TCP friendly bandwidth share $r^i_{l\_\mathrm{tcp}}$ (in bytes/s) is estimated with the use of the Eq. (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,

$$r^i_{l\_\mathrm{tcp}} = \frac{P}{t^{r-i}_{\mathrm{RTT}}\sqrt{\frac{2l_i}{3}} + 4t^{r-i}_{\mathrm{RTT}}\min\left(1, 3\sqrt{\frac{3l_i}{8}}\right) l_i(1+32l_i^2)}. \tag{2}$$

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

$$r^i_{l\_\mathrm{tcp}} = r^i_{l\_\mathrm{tcp}} + \frac{T_{\mathrm{rr}}}{(t^{r-i}_{\mathrm{RTT}})^2}P. \tag{3}$$

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

$$r^i = \min(r^i_{r\_\mathrm{tcp}}, r^i_{\mathrm{AIMD}}, r^i_{l\_\mathrm{tcp}}). \tag{4}$$

In the SRAMT-S variation, each time one receiver manager receives a receiver report informs the synchronisation manager to update the transmission rate of the sender streams. In order to update the transmission rate each stream, synchronisation manager polls the preferred transmission rates of all the receiver managers that correspond to receivers receiving this stream and sets the transmission rate $r_{\mathrm{stream}-j}$ of that stream to be the minimum preferred transmission rate of all the receivers receiving this stream,

$$\begin{aligned} r_{\mathrm{stream}-j} = {} & \min(r^i), \\ & \text{for all receivers } i \text{ listening to stream } j \\ & \text{(repeat this for all stream } 1 \ldots n). \end{aligned} \tag{5}$$

In the SRAMT-LE variation, 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 $r_{\mathrm{layer}-1}$ 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 $r_{\mathrm{layer}-2}$ the minimum values of r^i minus the $r_{\mathrm{layer}-1}$. This procedure repeats for all the layers,

$$\begin{aligned} r_{\mathrm{layer}-1} &= \min(r^i) \\ &\quad \text{for all receiver } i \text{ listening up to layer 1} \\ &\quad \text{(basic layer)}, \\ r_{\mathrm{layer}-2} &= \min(r^i) - r_{rmlayer-1} \\ &\quad \text{for all receiver } i \text{ listening up to layer 2}, \\ &\ldots \\ r_{\mathrm{layer}-n} &= \min(r^i) - r_{\mathrm{layer}-n-1} \\ &\quad \text{for all receiver } i \text{ listening up to layer } n. \end{aligned} \tag{6}$$

With the use of the above procedures, we ensure that sender will transmit TCP friendly traffic and in addition, due to the fact that the transmission rate of the first stream (SRAMT-S) and basic layer (SRAMT-LE) is set to the minimum value of receiver preferred transmission rates, SRAMT 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. [2]

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

3.3. 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 data based on RTP packets sequence numbers.
- 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 and estimates a TCP friendly bandwidth share $r_{r\_\mathrm{tcp}}^i$ every RTT time using Eq. (1). If the receiver experiences packet losses is using the following equation in order to estimate a TCP friendly bandwidth share (in bytes/s),

$$r_{r\_\mathrm{tcp}}^i = \frac{P}{t_{\mathrm{RTT}}^{e-i}\sqrt{\frac{2l_i}{3}} + 4t_{\mathrm{RTT}}^{e-i}\min\left(1, 3\sqrt{\frac{3l_i}{8}}\right)l_i(1+32l_i^2)}. \tag{7}$$

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

$$r_{r\_\mathrm{tcp}}^i = r_{r\_\mathrm{tcp}}^i + \frac{1}{t_{\mathrm{RTT}}^{e-i}}P. \tag{8}$$

Each time the receiver sends a receiver report to the sender includes the average value of $r_{r\_\mathrm{tcp}}^i$ since last receiver report.

In addition, the receiver has the capability to change streams (SRAMT-S) or add and remove layers (SRAMT-LE) based on the information that gathers itself and the information that sender includes in to RTP packets. The receivers' layer subscription or stream changes are synchronized at the end of a specific time period T_{epoch}, which we call epoch.

In the SRAMT-S variation, there are two cases that will lead to a receiver's transition towards another stream (we assume that the receivers have been informed about the upper and lower limits of each stream during the setup of the connection):

- If the multicast stream from which the receiver is currently receiving video has already reached its lowest (or highest) transmission rate and the receivers TCP friendly bandwidth share estimation is less (or more) than the stream transmission rate, then the receiver stops listening to this stream and joins the stream of a lower quality stream (or higher quality stream), if such a stream exists.
- When a receiver that co-exists in a stream with low (or high) capacity receivers but is preferring better (or worse) quality video, so it has been unable to increase (or decrease) the video quality of the current stream. The mechanism used aims in making the SRAMT-S more conservative and operates by counting the number of consecutive epochs the receiver's TCP friendly bandwidth share estimation was greater (or lesser) than the transmission rate of the multicast stream. When this number exceeds a certain limit (for our simulations this number was set to 4 which results a minimum of 20 s between stream change which is a time space enough in order to take a justified decision.), we assume

[2] 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.

that the receiver has indeed higher (or lower) capabilities and move it to a better (or worse) quality stream.

In the SRAMT-LE variation, the receivers 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:

$$r^i_{r\_tcp} \leqslant \sum_{j=1}^{k} r_{layer-j}. \tag{9}$$

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

During the evaluation of SRAMT stream/layer change mechanism we come to the conclusion that there is a trade off issue between protocol stability and protocol efficiency. If we allow more often stream/layer changes we improve protocol efficiency but the protocol is not so stable and the opposite. In order to overcome the above trade off we introduce the mechanism which can be used in order to avoid the unsuccessful stream/layer changes. With the use of that mechanism we change dynamically the time space between stream/layer changes in order to increase protocol stability but this has negative effects to protocol efficiency for some receivers. Actually we select the above approach because we believe that SRAMT stability is more important.

4. SRAMT parameters estimation

4.1. Packet loss rate estimation

Each receiver measures the packet loss rate based on RTP packets sequence numbers in each stream/layer (each stream/layer transmitted by the sender 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 [23] and has been evaluate and gives a good estimation of packet loss rate),

$$l_{i,l} = \frac{\sum_{j=0}^{m-1} w_j l_{i,l}^{m-j}}{\sum_{i=0}^{m-i} w_i} \quad \text{for receiver } i \text{ and stream/layer } l, \tag{10}$$

where $l_{i,l}$ is the smooth value of packet loss rate for stream/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: $\{1, 1, 1, 1, 0.8, 0.6, 0.4, 0.2\}$. In the SRAMT-S variation, the receiver reports as packet loss rate l_i the packet loss rate $l_{i,l}$ of the stream l, which receives. In the SRAMT-LE variation, the receiver estimates the packet loss rate l_i, for all the layers $(1 \ldots k)$ that the receiver receives, with the following equation:

$$l_i = \frac{\sum_{j=1}^{k} l_{i,j} * r_{layer-j}}{\sum_{j=1}^{k} r_{layer-j}}. \tag{11}$$

4.2. RTT estimations

When a receiver i receives a RTP packet from a sender stream/layer, uses the following algorithm in order to estimate the 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 ($T_{\text{timestamp}}$) and the local time that receives that packet (T_{receiver}) in order to estimate the one way delay form sender to receiver (T_{oneway}),

$$T_{\text{oneway}} = T_{\text{receiver}} - T_{\text{timestamp}}. \tag{12}$$

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}

$$t_{\text{RTT}} = 2T_{\text{oneway}}. \tag{13}$$

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 result in order to get accurate RTT estimations (t^{e-l}_{RTT}), receivers have to take the above assumptions into account. For this reason, we use a parameter a and we can write the Eq. (13) as

$$t^{e-l}_{\text{RTT}} = (1 + a)T_{\text{oneway}}. \tag{14}$$

The parameter a is used in order to smooth the estimation of the RTT due to the potential unsynchronised 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^{e-l}_{RTT} values through a filter similar to the filter, which they use for filtering the values of packet loss rate.

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):

$$t^{r-i}_{\text{RTT}} = A - t_{\text{LSR}} - t_{\text{DLSR}}. \tag{15}$$

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 all the streams (SRAMT-S) or the basic layer (SRAMT-LE).

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

$$a = \frac{t^{r-i}_{\text{RTT}}}{T_{\text{oneway}}} - 1. \tag{16}$$

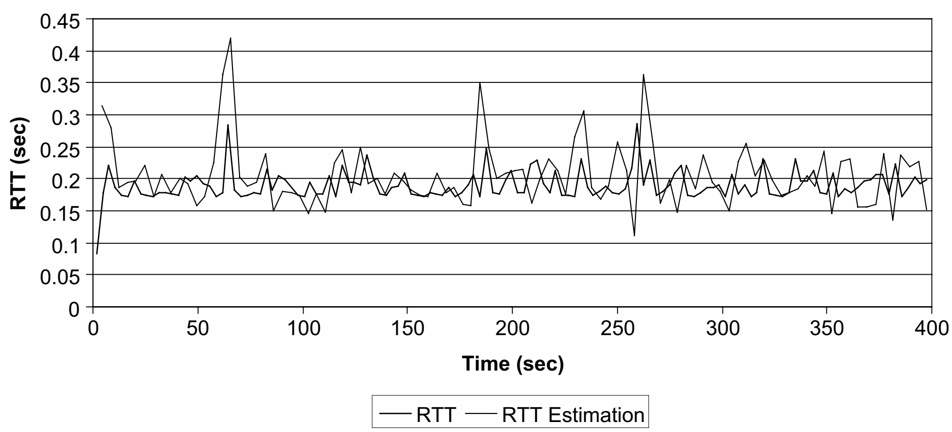

Fig. 2. RTT estimations.

Fig. 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 over a 1 Mbit 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 gives 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 stream/layer l. In the SRAMT-S variation, the receiver is using as RTT estimation t_{RTT}^{e-i} the t_{RTT}^{e-l} of the stream l which receives. In the SRAMT-LE variation, the receiver is using for TCP friendly transmission rate estimation the average value of t_{RTT}^{e-l} for all the layers $(1 \ldots k)$ that receives:

$$t_{\mathrm{RTT}}^{e-i} = \frac{\sum_{l=1}^{k} t_{\mathrm{RTT}}^{e-l}}{k}. \tag{17}$$

5. Extensions to RTP/RTCP

As we have already mentioned, the operation of SRAMT 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 uses the extension mechanism of RTP in order to add the following fields in to RTP header:

- T_{epoch}: The specific time period called epoch, in which the receivers have the capability to change streams (SRAMT-S) or changes their layer subscription level (SRAMT-LE).
- 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 streams $r_{\mathrm{stream}-j}$, $j = 1, \ldots, n$ or sender layers $r_{\mathrm{layer}-j}$, $j = 1, \ldots, n$.
- End of epoch flag: This flag is used in order the receiver to be informed about the end of an epoch and synchronize their stream/layer changes.

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 reports an application specific part, which contains the average value of their estimations for TCP friendly bandwidth share $r_{r\_\mathrm{tcp}}^{i}$ and the packet loss rate estimation l_i, since last receiver report. Moreover the receivers add to their receiver report, the stream number that receive (SRAMT-S) or the current layer subscription (the maximum layer up to which the receives listening) k (SRAMT-LE).

In the SRAMT-S variation the above described extensions to RTP/RTCP are used to all streams. In the SRAMT-LE variation 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.

6. 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 multicast stream, the transmission of the multicast stream will stop only if that receiver was the only one receiver listening to that multicast stream behind that node. In addition, if two receivers join different streams/layers at the same time, the receiver which joins the streams/layer with the lower transmission rate might observe losses that were not caused by his action but by the action of receiver join the stream/layer with the higher transmission rate.

Similar research has shown [13] that, if the receivers synchronize their stream/layer changes, the above problems can be minimized. For this reason the receivers' stream/layer changes are synchronized in the end of each epoch. The sender marks the next RTP packet in all streams (in SRAMT-S variation) or only to the basic layer (in SRAMT-LE where 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_{epoch} in all the RTP packets that transmits. Receivers can change streams/layers either when receive a special marked packet or after ($T_{\text{epoch}} + T_{\text{oneway}}$) 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_{epoch} to be 5 s in order to allow receivers to quickly find the stream or subscription level which fulfils better their requirements. The small value of T_{epoch} does not cause problems due to the tracing and suspending of unsuccessful stream/layer changes mechanism that SRAMT supports.

7. Scalability issues

The RTCP adaptive transmission mechanism defines as 5 s 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 [17] 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 $[0, T_{\text{rand}}]$ with density of

$$T_{\text{wait}} = \begin{cases} \frac{1}{\exp^{\lambda}-1} * \frac{\lambda}{T_{\text{rand}}} \exp^{(\lambda/T_{\text{rand}})^z} \\ \quad \text{if } 0 \leqslant z \leqslant T_{\text{rand}}, \\ 0 \quad \text{otherwise.} \end{cases} \tag{18}$$

Each receiver schedules the RTCP retransmission timeout to be T_{wait}. If the receiver receives a receiver report from an other receiver with TCP friendly bandwidth share $r^i_{r\_\text{tcp}}$ 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 [17] shows analytically, with the appropriate selection of the Eq. (18) parameters (λ, 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 s 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.

The fact the partial suppressed method does not affect the stream/layer change mechanism and the opposite has as result good scalability performance of the SRAMT protocol.

In addition the partial suppressed method does not change the architecture of the SRAMT sender entity (actually the partial suppressed method is implemented only to SRAMT receiver entity) due to the fact that when the partial suppressed method is in use the SRAMT sender entity store information only for a part of the receivers. The only drawback that we have mentioned is an overhead in SRAMT sender entity processing due to the fact that the group of receiver for which the SRAMT sender entity stores information chance more often.

8. Mechanism evaluation

In this section, we present a number of simulations that we run in order to analyse the behavior of SRAMT, during the multicast transmission of multimedia data. We implemented SRAMT and run simulations in the LBNL network simulator ns-2 [15]. We evaluate both SRAMT-S and SRAMT-LE variations.

8.1. Heterogeneous multicast environments—TCP friendliness

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

Fig. 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, they have delay 10 ms and they use the RED (Random Early Drop—[9]) 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 to a group of 6 receivers (R1–R6) with different capabilities. In addition we have 3 TCP sources TCP1, TCP2, TCP3, which transmit data to Sink1, Sink2, Sink3 respectively. We model the TCP sources as "4.3BSD Tahoe TCP" [22] 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 SRAMT and a TCP connection with the same RTT time as the sender/receiver pair. We run this simulation two times, one with the use of SRAMT-S variation and one with the use of SRAMT-LE variation.

In the SRAMT-S variation the sender transmits three multicast streams with the following limits: stream one: 50–200 Kbps, stream two: 200–

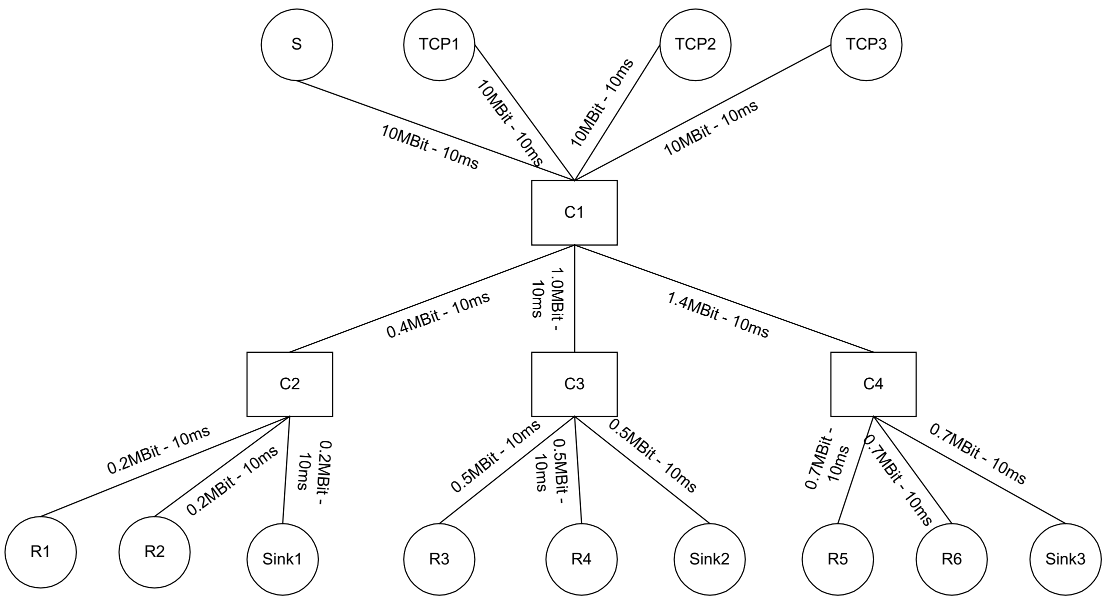

Fig. 3. Topology with no share links.

600 Kbps and stream three: 600–1000 Kbps. We execute this simulation for 1000 s and the sender starts transmitting the stream one with transmission rate 50 Kbps, the stream two with transmission rate 200 Kbps and the stream three with transmission rate: 600 Kbps. Receivers R1 and R2 join the stream one, receivers R3 and R4 join the stream two and Receivers R5 and R6 join the stream three. Figs. 4–6 shows the bandwidth distribution to bottleneck links C1–C2, C1–C3 and C1–C4, respectively.

In the SRAMT-LE variation the sender transmits three layers with the following limits: layer one (basic layer): 50–200 Kbps, layer two: 50–400 Kbps and layer three: 50–400 Kbps. With this configuration the maximum cumulative transmission rate up to layer one is 200, up to layer two is 600 Kbps and up to layer three is 1000 Kbps. We execute this simulation for 1000 s and the sender starts transmitting all the layers with transmission rate 150 Kbps. 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). Figs. 7–9 shows the bandwidth distribution to bottleneck links C1–C2, C1–C3 and C1–C4, respectively.

As the above figures suggest, receivers behave as we except in both variation of SRAMT. These figures indicate that SRAMT is in general fair towards to TCP connections and treats the heterogeneous group of the receivers with fairness. In all bottleneck links SRAMT behaves as is expected, and shares the available bandwidth with the TCP connection with the same RTT delay. The behavior of SRAMT ("seeking" for available bandwidth

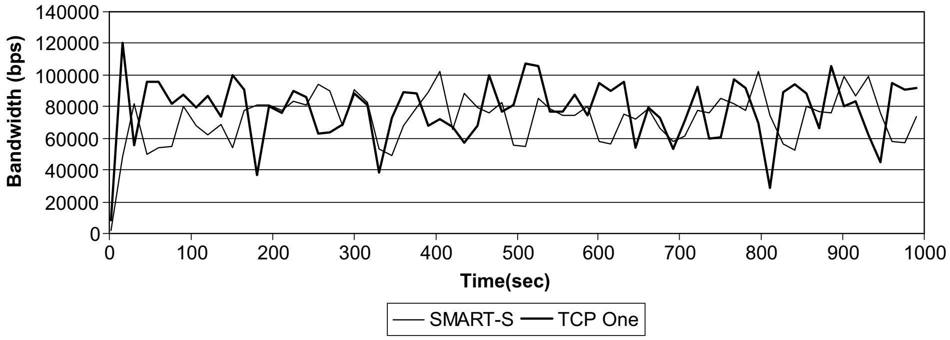

Fig. 4. Bandwidth distribution on C1–C2 bottleneck link with the use of SRAMT-S.

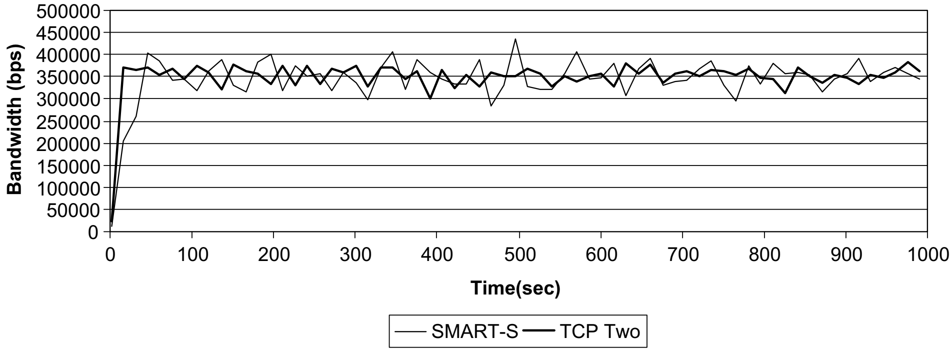

Fig. 5. Bandwidth distribution on C1–C3 bottleneck link with the use of SRAMT-S.

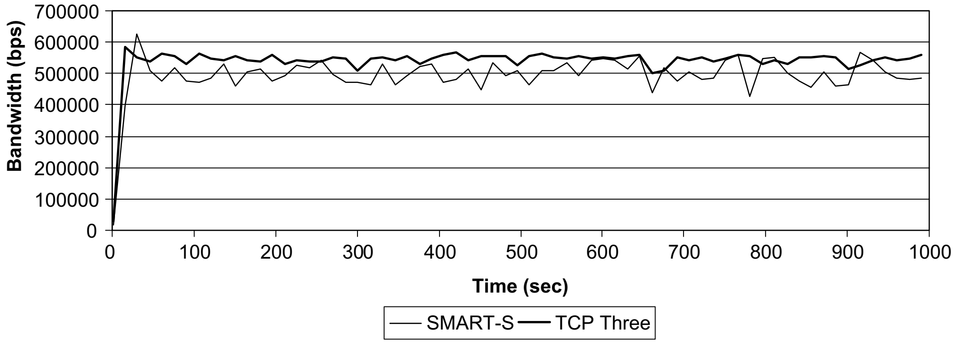

Fig. 6. Bandwidth distribution on C1–C4 bottleneck link with the use of SRAMT-S.

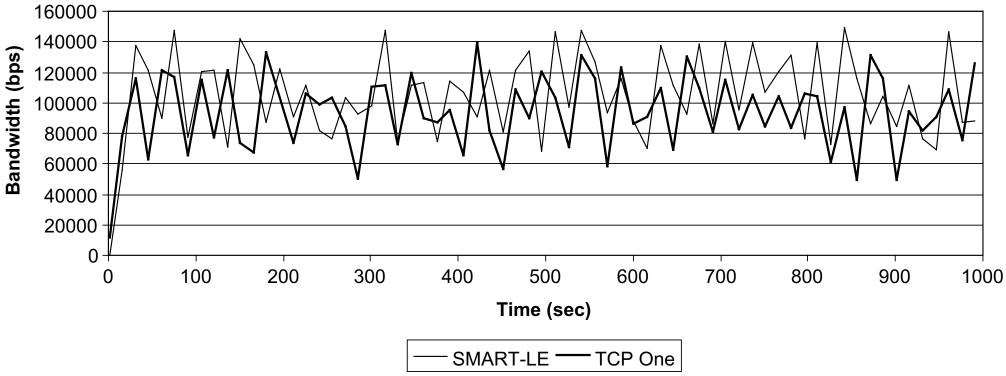

Fig. 7. Bandwidth distribution on C1–C2 bottleneck link with the use of SRAMT-LE.

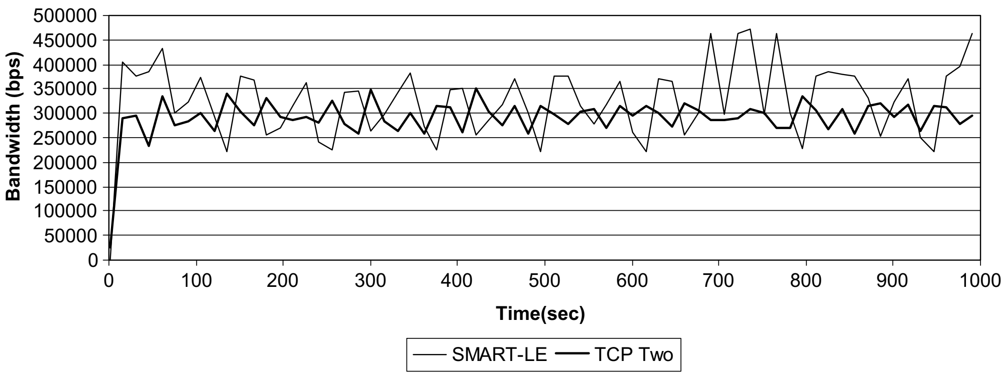

Fig. 8. Bandwidth distribution on C1–C3 bottleneck link with the use of SRAMT-LE.

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 and the TCP flows get approximately the same bandwidth share of the bottleneck links. In addition both SRAMT-S and

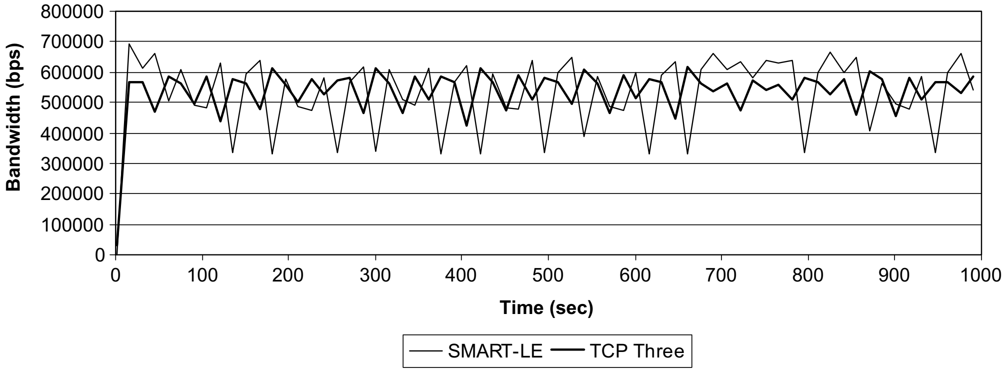

Fig. 9. Bandwidth distribution on C1–C4 bottleneck link with the use of SRAMT-LE.

SRAMT-LE have similar behavior and performance mainly due to the fact that the simulation topology does not have any link shared among multicast stream. The only difference, which we mention between the SRAMT-S variation and SRAMT-LE variation is that the SRAMT-S variation keeps the transmission rates of the streams more invariable than SRAMT-LE variation keeps the transmission rates of the layers.

8.2. Multicast environments with share links

In this simulation, we investigate the performance of SRAMT 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, when the actions of one receiver affect other receivers.

Fig. 10 shows the topology of this simulation. The bandwidth of each link is given to the simulation topology and varies from 0.2–10.0 Mbps. All

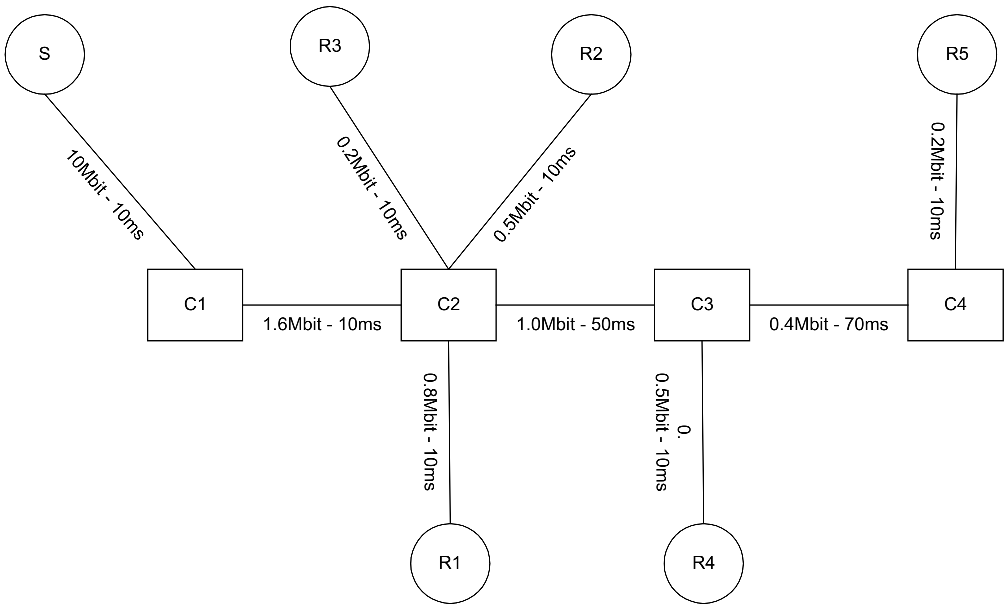

Fig. 10. Topology with share links.

the links in the simulation topology are full duplex, have delay, which varies from 10–70 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 to a group of 5 receivers (R1–R5) with different capabilities. 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 s. As [19] suggests the above traffic generator models background web traffic. We run this simulation two times, one with the use of SRAMT-S variation and one with the use of SRAMT-LE variation and we execute both simulations for 1000 s. In both simulations the configuration of the SRAMT-S and SRAMT-LE was the same with the previous simulations, which is presented in Section 8.1. In order to avoid synchronization, the receivers join randomly the stream one (SRAMT-S) or the layer one (SRAMT-LE) during the first 3 seconds of the simulation. Fig. 12 shows the bandwidth share of the receivers R1–R5 during SRAMT-S simulation and Fig. 11 shows the bandwidth share of the receivers R1–R5 during SRAMT-LE simulation.

In the case of SRAMT-LE, with the above describe topology, we expect that receivers R5 and

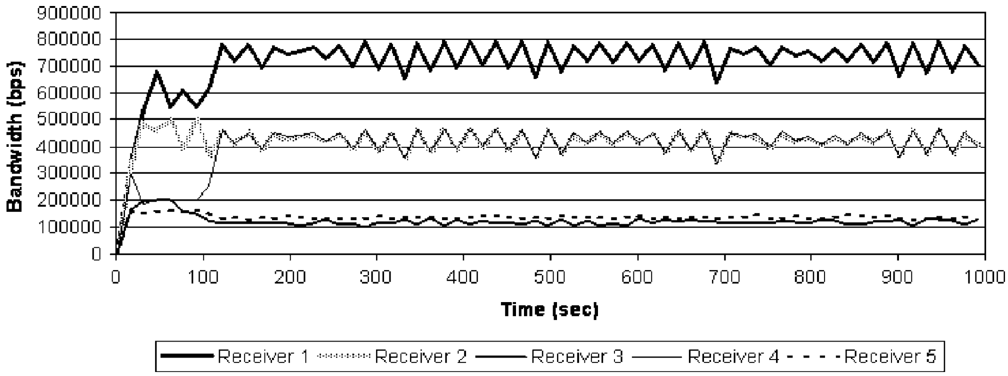

Fig. 11. Bandwidth shares of Receiver R1 to Receiver R5 with the use of SRAMT-LE.

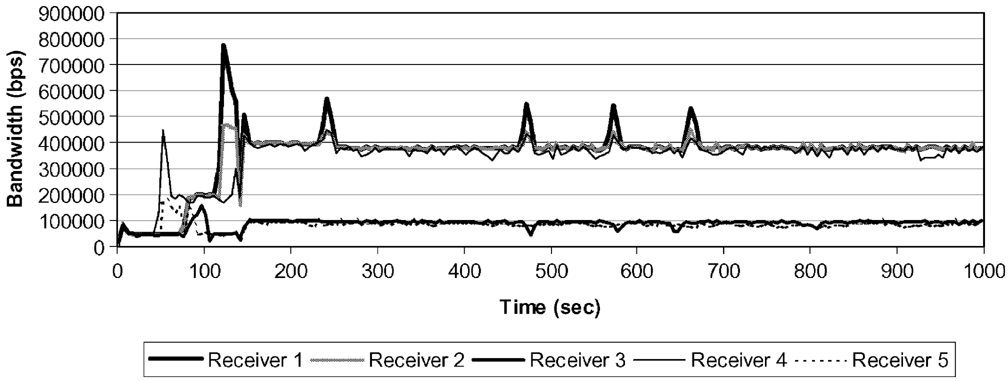

Fig. 12. Bandwidth shares of Receiver R1 to Receiver R5 with the use of SRAMT-S.

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 receiver R1 will join up to layer three (layer subscription level 3). As Fig. 11 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 level one which is close to 120 Kbps and not close to 200 Kbps, which is the expected transmission rate, based on the topology of Fig. 10. 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 s) have join the layers which fulfils 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.

In the case of SRAMT-S, the topology of Fig. 10 restricts the performance of the SRAMT-S, due to the fact that the available bandwidth of link C1–C2 is not enough for the transmission of all the sender streams and the available bandwidth of link C2–C3 is not enough for the transmission of sender streams one and two. As Fig. 12 suggests the SRAMT-S has lower performance than the SRAMT-LE. In addition as Fig. 12 shows, receivers R1, R2 and R4 join the stream two of sender and receiver R3 and R5 join the stream one of the sender. Moreover the receiver R1 tries to join the sender stream three but returns immediately to stream two, due to congestion, until the end of the simulation. It is obvious that the topology of Fig. 10 restricts the performance of the SRAMT-S because of the bandwidth of links C1–C2 and C2–C3.

In order to evaluate the SRAMT-S in a more "friendly" topology, we increase the bandwidth of link C1–C2 to 3.0 Mbps and the bandwidth of link C2–C3 to 1.4 Mbps and we run again the simulation for SRAMT-S (again the background traffic consumes maximally the 50% of the C1–C2 and C2–C3 links capacity). With this change in the topology of Fig. 10, we expect that receiver R1 will join the sender stream three, receivers R2 and R4 will join the sender stream two and R3 and R5 will join the sender stream one. Fig. 13 shows the results of this simulation. As this figure suggests after some seconds each receiver joins the stream, which we expect and receives also a bandwidth share close to the bandwidth share, which we expect. The receivers have join the sender stream which fulfils better their capabilities after some

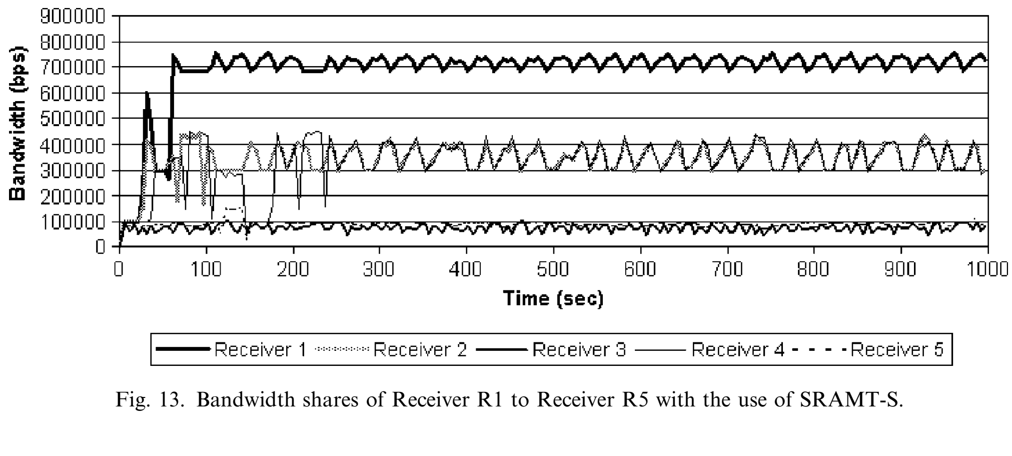

Fig. 13. Bandwidth shares of Receiver R1 to Receiver R5 with the use of SRAMT-S.

unsuccessful stream changes (during the first 200 seconds) and stay at that stream until the end of the simulation (due to the tracing of unsuccessful stream changes that SRAMT-S offers). Again the transmission rate of stream one is close to 100 Kbps and not close to 200 Kbps due to the fact that stream one is the stream 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). As result the estimations of TCP friendly bandwidth share of receiver R5 is low and the sender keeps the transmission rate of stream one low in order to service also the receiver R5. In addition, due to the synchronization of stream changes the undesirable problems are minimal and in general the receivers actions does affect the bandwidth shares of the other receivers.

The most common queuing management technique in the Internet today is the droptail queue management (mostly because its easy implementation). In order to evaluate the performance of the SRAMT mechanism when the droptail queue management technique is used, we change the queuing management technique of the routers of the topology of Fig. 10 to droptail and we run again the simulation using the SRAMT-S variation of the SRAMT mechanism. Also in this simulation, we increase the bandwidth of link C1–C2 to 3.0 Mbps and the bandwidth of link C2–C3 to 1.4 Mbps and again the background traffic consumes maximally the 50% of the C1–C2 and C2–C3 links capacity. Fig. 14 shows the results of this simulation. As Fig. 14 shows, SRAMT has similar behavior when droptail queue management technique is used and the receivers after some time have join the sender stream, which fulfils better their capabilities. The receivers stay at the appropriate stream until the end of the simulation (due to the tracing of unsuccessful stream changes that SRAMT offers). The only drawback, which we mentioned when we used the droptail queue management technique is the fact that the bandwidth share of the SRAMT is reduced (comparing the bandwidth share which receives which we use RED queue management technique). This phenomenon affects mainly the receivers of the low bit rate streams. The above behavior is expected and can be justified as follows: the droptail queue management technique does not "protect" from synchronization among the streams and in addition droptail queue management technique results in more packet losses in the routers (due to queue overflow) comparing with RED queue management technique. This has as result higher packet losses and lower transmission rates for the SRAMT mechanism. The streams that are affected the most are the streams with the smaller transmission rates due to the fact that these streams pass through more droptail queues based on the Fig. 14 topology.

8.3. Comparison of SRAMT-S and SRAMT-LE during the evaluation

General the behavior of SRAMT-S variation and SRAMT-LE variation is the same. In this section we describe some differences, which we men-

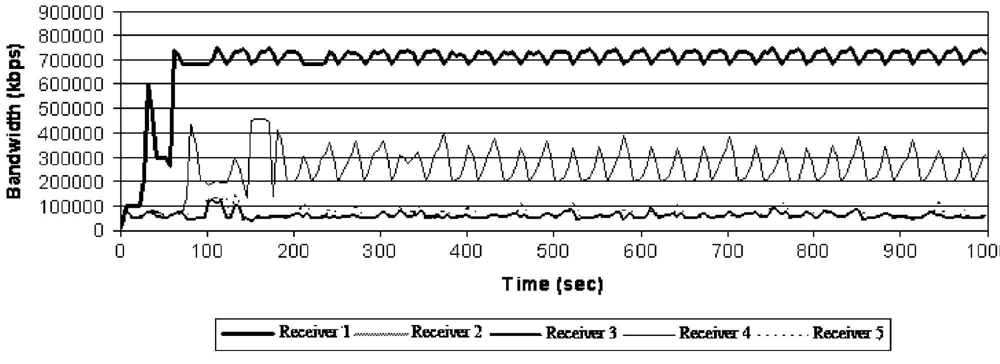

Fig. 14. Bandwidth shares of Receiver R1 to Receiver R5 with the use of SRAMT-S and droptail queue management.

tioned between the SRAMT-S variation and SRAMT-LE variation during the performance evaluation. Mainly these differences derived from the different characteristics of the simulcast and layered encoding approaches.

One main difference between SRAMT-S and SRAMT-LE is the better performance of SRAMT-LE in topologies with a multicast distribution tree that is shared among the receivers. In these topologies the SRAMT-LE behaves better mainly due to the fact the simulcast approach, which SRAMT-S is using, wastes bandwidth by essentially duplicating the transmission of the content in multiple streams. On the other hand the layered encoding approach, which SRAMT-LE is using, wastes bandwidth as overhead for the operation of layered encoding and decoding of video information. Depending of the encoding which is used, the overhead of layered encoding may be more that 20%, which means that the layered encoding approach needs 20% more bandwidth comparing the simulcast approach in order to provide the same experience to the end user (In [12], authors tries to measure the above affect and compare simulcast and layered encoding). Moreover the implementation of layered encoders/decoders is more complex that the traditional encoders, which are used during the simulcast, approach.

One other difference between SRAMT-S and SRAMT-LE is the fact that the SRAMT-S variation keeps the transmission rates of the streams more invariable than SRAMT-LE variation keeps the transmission rates of the layers. This behavior derives from the fact that with the use of SRAMT-LE variation, the bottleneck links are shared among more multicast streams than the use SRAMT-S variation. For example in Fig. 3 the C1–C4 bottleneck link is shared between two streams (TCP three and sender stream three) when we use the SRAMT-S variation and is shared between four streams (TCP three and sender layer one, two and three) when we use SRAMT-LE variation.

9. Comparison of SRAMT with other schemes

In this section we compare the performance of SRAMT-S and SRAMT-LE mechanisms with other mechanisms 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 congestion control scheme.

We compare the SRAMT-LE with the following layered encoding schemes:

- PLM [13]: 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 [21]: 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 [23]: 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.

We compare the SRAMT-S with the following simulcast schemes:

- DSG [6]: DSG (Destination Set Grouping) is an end to end mechanism for the simulcast transmission of multimedia data which bases its estimations on the current network conditions. DSG uses the packet loss rate in order to estimate the network conditions. The receivers send their estimations regarding the network condition to the sender using a special mechanism in order to avoid the feedback implosion problem. Receivers decide for the multimedia stream, which they will receive based on the information, which they collect, and the information, which they receive from the sender. Also the receivers' changes among the streams are synchronized.
- IRFM [11]: IRFM (Inter-Receiver Fair Multicast) is an end to end mechanism for the simulcast transmission of multimedia data. IRFM uses the packet loss rate in order to estimate the network conditions and also uses a fairness function in order to serve the receivers with fairness. Main characteristic of IRFM is that it uses only two multimedia streams, one for the low capabilities receivers and one for the high capabilities receivers.

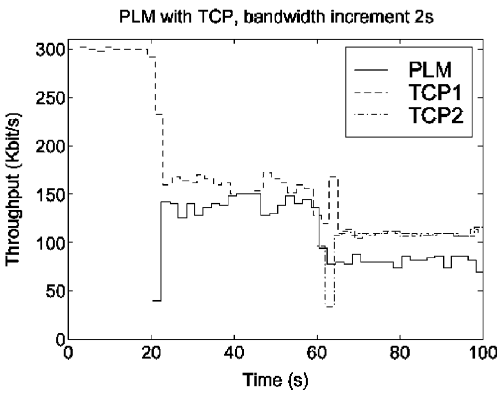

Fig. 15. PLM performance against TCP traffic.

Fig. 15 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 Fig. 2 shows, the PLM session adapts all most perfectly to the available bandwidth in presence of TCP flows. Comparing the PLM behavior with the SRAMT-LE behavior we can draw the following conclusions: PLM has more stable transmission rate and change its transmission rate in steps comparing with SRAMT-LE which can not keep its transmission rate stable and changed it continues during the entire experiment (Fig. 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 SRAMT-LE, TCP and SRAMT-LE traffics are share almost them equally the available bandwidth. In order to summarize, both PLM and SRAMT-LE have good behavior against the TCP traffic with PLM offering a more stable transmission rate and SRAMT-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, it is 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.

Fig. 16 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 Fig. 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

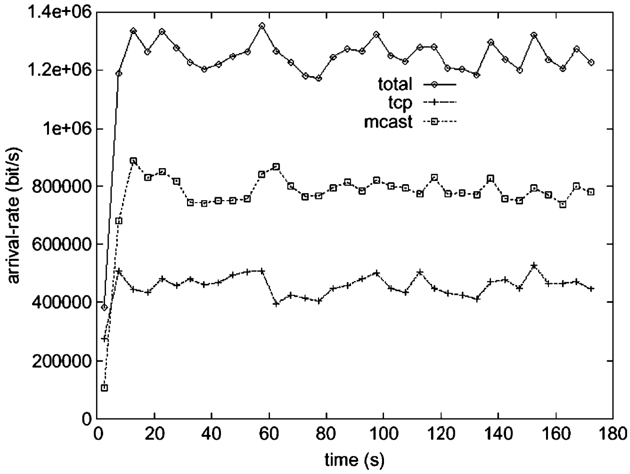

Fig. 16. RLC performance against TCP traffic.

does not. On the other hand, TCP and RLC do not starve each other when competing. Comparing the RLC behavior with the SRAMT-LE behavior we can draw the following conclusions: both RLC and SRAMT-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 SRAMT-LE have similar convergence times to the stable state. In order to summarize, SRAMT-LE has better behavior against the TCP traffic comparing with RLC and this is because the TCP analytical model used by SRAMT-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 SRAMT-LE.

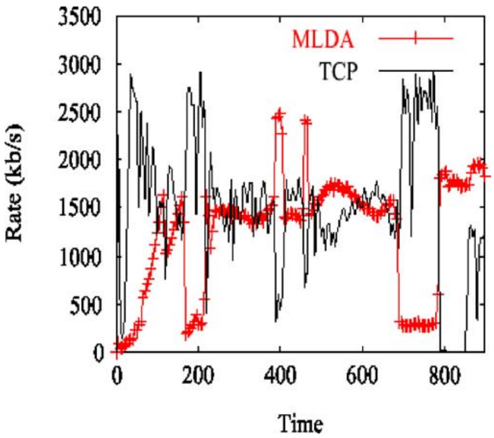

Fig. 17. MLDA performance against TCP traffic.

Fig. 17 shows how the MLDA shares a bottleneck link with TCP traffic. Fig. 17 shows the bandwidth share between MLDA and TCP traffic in the bottleneck link. As Fig. 17 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 SRAMT-LE behavior we can draw the following conclusions: the SRAMT-LE behavior is friendlier that MLDA behavior against TCP traffic mainly due to the fact the SRAMT-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 SRAMT-LE. Moreover, both MLDA and SRAMT-LE do not keep their transmission rates stable but they have fluctuation on their transmission rates. In order to summarize, both MLDA and SRAMT-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
Comparison of SRAMT-LE with the other layered encoding schemes

Parameter/mechanism	SRAMT-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
Scalability	Well—partial suppression method	Well—not require feedback for the client	Well—not require feedback for the client	Well partial suppression method
Limitations	No	Fair queuing mechanism in routers	No	No

Table 1 summarizes the comparison of SRAMT-LE against the others layered encoding schemes. As this table shows, SRAMT-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 SRAMT-LE mechanism is the fact that SRAMT-LE has fluctuation on its transmission rate and does 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 SRAMT-LE behavior.

Fig. 18 shows the transmission of multimedia data with the use of DSG. The scenario of this experiment includes the transmission of three multicast streams over the Internet. As Fig. 18 shows, DSG has a relative stable operation and its transmission rate does not have heavy fluctuations. In addition, the time until the DSG obtains stable operation is satisfactory. Regarding TCP friendliness, paper [6] does not provide any information. Comparing DSG with SRAMT-S we can draw the following conclusions: DSG has better performance than SRAMT-S regarding stability and both SRAMT-S and DSG have satisfactory performance regarding the time to obtain stable operation. Moreover SRAMT-S offers a TCP friendly operation.

Fig. 19 shows the transmission of multimedia data with the use of IRFM together with TCP traf-

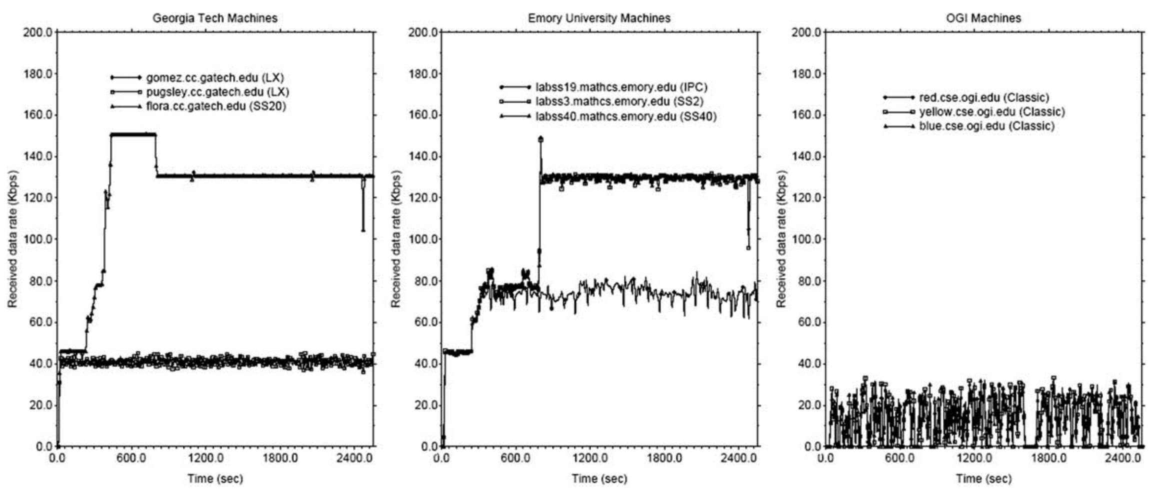

Fig. 18. DSG performance.

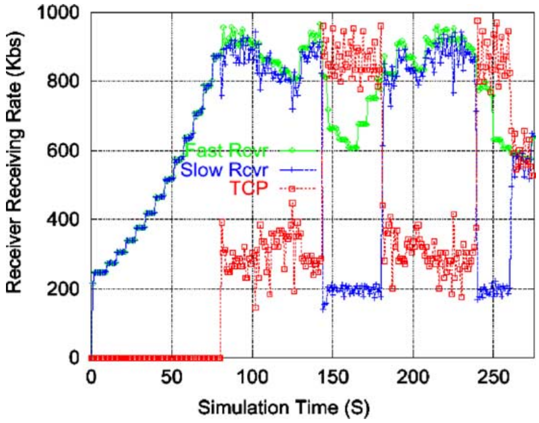

Fig. 19. IRFM performance against TCP traffic.

fic. The scenario of this experiment includes the transmission of one TCP connection and IRFM traffic (two multicast streams) over a network link. As Fig. 19 shows, IRFM has friendly behavior against TCP traffic (by the meaning that TCP traffic does not starve) and some times IRFM gets more bandwidth than TCP and some times TCP gets more bandwidth than IRFM. In addition IRFM has relative stable operation but has some heavy fluctuations during the transmission of the multimedia data. Regarding scalability issues, IRFM is using a feedback suppression mechanism during the transmission of feedback information from the receivers to the sender. Comparing IRFM with SRAMT-S we can draw the following conclusions: SRAMT-S has better performance than IRFM regarding TCP friendliness and it also needs less time to obtain stable operations. Regarding stability both IRFM and SRAMT-S have similar performance.

Table 2 summarizes the comparison of SRAMT-S against the others simulcast schemes. As this table shows, SRAMT-S has good performance against TCP traffic and in general terms has good performance comparing with the other simulcast schemes. The main drawback of the SRAMT-S mechanism is the fact that SRAMT-S has fluctuation on its transmission rate and does not keep its transmission rate stable.

10. Future work

Our future work includes the investigation of the fluctuations in SRAMT transmission rate in order the SRAMT to transmit more smooth transmission rates and provide a better experience to the end users. In addition our future work includes the investigation of dynamically adding more streams/layers instead of the static number of streams/layers that SRAMT supports now. This will provide more flexibility to the operation of the SRAMT mechanism. Moreover we plan to implement a prototype of SRAMT (for both variations) 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. In addition we will perform a detail validation of SRAMT through test over the Internet with use of large participants groups and we will investigate the scalability of proposed mechanism and how the proposed mechanism will deal with the feedback implosion problem. Furthermore we will compare the SRAMT-S with SRAMT-LE based on the end user experience and we will measure the bandwidth overhead in both SRAMT variations (in the SRAMT-S varia-

Table 2
Comparison of SRAMT-S with the other simulcast schemes

Parameter/mechanism	SRAMT-S	DSG	IRFM
TCP friendliness	Very good	–	Modest
Stable transmission rate	Modest	Yes	Modest
Convergence time	Satisfactory	Satisfactory	Modest
Stable operation	Yes	Yes	Modest
Scalability	Very good	Modest	Modest
Limitations	No	No	No

tion we have bandwidth overhead due to the redundant transmission of the same multimedia information and in the SRAMT-S variation we have bandwidth overhead due to the use of layered encoding). In addition we plan to investigate the fairness of SRAMT mechanism by performing details simulation and measuring SRAMT fairness versus various parameters (for example number of receivers, versus link capacity, link delay, number of streams) in order to receive more reliable results about SRAMT fairness. Finally we intend to enhance the proposed mechanism by adding a mechanism in order to dynamically choose and modify the parameters that regulate the aggressiveness of the adaptation.

11. Conclusion

In this paper, we present the behavior investigation of the SRAMT, a mechanism for multicast transmission of adaptive multimedia data in a heterogeneous group of receivers. SRAMT 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. The proposed mechanism uses RTP/RTCP protocols for the transmission of multimedia data. We are concentrating on the design of a mechanism for monitoring the network condition and estimating the appropriate rate for multicast transmission of multimedia data in order to treat with fairness the clients. We propose two variations of SRAMT: (1) SRAMT-Simulcast (SRAMT-S) which is using simulcast approach for the transmission of multicast data and (2) SRAMT-Layered Encoding (SRAMT-LE) which is using layered encoding approach for the transmission of multicast data. We investigate the behavior of SRAMT through a number of simulations. Main conclusion of the simulation was that SRAMT has friendly behavior against the dominant traffic types (TCP traffic) of today's Internet and good behavior during congestion condition in both of its versions. In addition SRAMT treats with fairness a heterogeneous group of receivers.

We compare also the behavior of SRAMT with other schemes available to the literature and we come to the conclusion that SRAMT- provides good performance comparing with other schemes available to the literature.

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, 17–19 October 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), 9–12 October 2001.

[5] R. Braden, D. Clark, S. Shenker, Integrated services in the internet architecture: an overview, RFC 1633, 1994.

[6] S.Y. Cheung, M. Ammar, X. Li, On the use of destination set grouping to improve fairness in multicast video distribution, INFOCOM 96, San Fransisco, March 1996.

[7] S. Deering, R. Hinden, Internet Protocol, Version 6 (IPv6) specification, RFC 2460, 1998.

[8] 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.

[9] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking 1 (4) (1993) 397–413.

[10] S. Floyd, K. Fall, Promoting the use of end-to-end congestion control in the Internet, IEEE/ACM Transactions on Networking 7 (4) (1999) 458–472.

[11] T. Jiang, E.W. Zegura, M. Ammar, Inter-receiver fair multicast communication over the Internet, in: Proceedings of the 9th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), June 1999, pp. 103–114.

[12] T. Kim, M.H. Ammar, A comparison of layering and stream replication video multicast schemes, in: Proceedings of the NOSSDAV'01, Port Jefferson, NY, 25–26 June 2001.

[13] A. Legout, E. Biersack. PLM: fast convergence for cumulative layered multicast transmission schemes, in:

Proceedings of ACM SIGMETRICS'2000, Santa Clara, CA, USA, June 2000.
[14] S. McCanne, V. Jacobson, Receiver-driven layered multicast, 1996 ACM Sigcomm Conference, August 1996, pp. 117–130.
[15] S. McCanne, S. Floyd, The UCB/LBNL network simulator, software online. http://www.isi.edu/nsnam/ns/.
[16] K. Nichols, S. Blake, F. Baker, D. Black, Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers, RFC 2474, 1998.
[17] J. Nonnenmacher, E.W. Biersack, Optimal multicast feedback, in: Proceedings of the Conference on Computer Communications (IEEE Infocom), San Francisco, USA, March 1998.
[18] J. Pandhye, J. Kurose, D. Towsley, R. Koodli, A model based TCP-friendly rate control protocol, in: Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Basking Ridge, NJ, June 1999.
[19] 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, October 1996, pp. 171–180.
[20] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: a transport protocol for real-time applications, RFC 1889, IETF, January 1996.
[21] 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.
[22] W. Stevens, TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms, RFC 2001, January 1997.
[23] L. Vicisiano, L. Rizzo, J. Crowcroft, TCP-like congestion control for layered multicast data transfer, in: IEEE INFOCOM, March 1998, pp. 996–1003.
[24] J. Widmer, M. Handley, Extending equation-based congestion control to multicast applications, in: ACM SIGCOMM, August 2001.

Christos Bouras obtained his Diploma and Ph.D. from the Computer Science and Engineering Department of Patras University (Greece). He is currently an Assistant Professor in the above department. Also he is a scientific advisor responsible of Research Unit 6 and director of Networking Technologies Sector, in Computer Technology Institute (CTI), Patras, Greece. His research interests include Networks and Protocols, Educational Technology, Telematics and New Services. He has extended professional experience in Design and Analysis of Networks, Protocols, Telematics and New Services, Distance Learning and Education, Electronic Publishing, High Speed Networks and Interactive Hypermedia and Multimedia. He has published 100 papers in various well-known refereed conferences and journals. He is a co-author of five books in greek. He has been a PC member and referee in various journals and conferences. He has participated in R&D projects such as RACE, ESPRIT, TELEMATICS, EDUCATIONAL MULTIMEDIA, ISPO, EMPLOYMENT, ADAPT, STRIDE, EUROFORM, IST, GROWTH and others. Also he is a member of the team of experts in the Greek Research and Technology Network (GRNET), ACM, IEEE, EDEN, AACE and New York Academy of Sciences.

Apostolos Gkamas obtained his Diploma, Master Degree and PhD from the Computer Engineering and Informatics Department of Patras University (Greece). He is currently an R&D Computer Engineer at the Research Unit 6 of the Computer Technology Institute, Patras, Greece. His research interests include Computer Networks, Telematics, Distributed Systems, Multimedia and Hypermedia. More particular he is engaged in the transmission of multimedia data over networks and multicast congestion control. He has published 8 papers in Journals and 26 papers in well-known refereed conferences. He has participated in R&D projects such as OSYDD, RTS-GUNET, ODL-UP, ODL-OTE, TODAY'S STORIES, ATMNet, "Technical Consultant to Pedagogical Institute" and ELECTRA and is currently involved in the project Teleteaching Service for Greek PTT, IPv6—GRNET and 6NET (LARGE-SCALE INTERNATIONAL IPv6 TESTBED—IST-2001-32603).