J Netw Syst Manage (2011) 19:143–177
DOI 10.1007/s10922-010-9159-8

Adaptive Smooth Simulcast Protocol (ASSP) for Video Applications: Description and Performance Evaluation

Christos Bouras · Apostolos Gkamas · Georgios Kioumourtzis

Published online: 27 February 2010

Abstract In this paper, we present Adaptive Smooth Simulcast Protocol (ASSP) for simulcast transmission of multimedia data over best-effort networks. ASSP is a new multiple-rate protocol that implements a single rate TCP-friendly protocol as the underlying congestion control mechanism for each simulcast stream. The key attributes of ASSP are: (a) TCP-friendly behavior, (b) adaptive per-stream transmission rates, (c) adaptive scalability to large sets of receivers and (d) smooth transmission rates that are suitable for multimedia applications. We evaluate the performance of ASSP under an integrated simulation environment which combines the measurements of both network and video performance metrics. We also compare ASSP against other proposed solutions and the results demonstrate that the performance of ASSP is significantly better than the tested solutions. Finally, ASSP is a practical solution with very low implementation complexity for video transmission over best-effort networks.

Keywords Multimedia transmission · TCP friendly · Media friendly · Congestion control · Multicast

C. Bouras (✉) · A. Gkamas · G. Kioumourtzis
Research Academic Computer Technology Institute and University of Patras, Patras, Greece
e-mail: bouras@cti.gr
URL: http://ru6.cti.gr/bouras

A. Gkamas
e-mail: gkamas@cti.gr
URL: http://ru6.cti.gr/gkamas

G. Kioumourtzis
e-mail: gkioumou@ceid.upatras.gr
URL: http://gkioumou.googlepages.com/home

1 Introduction

Video streaming has taken off the last few years, as there has been an increasing interest from the industry as well as the users' community. New services such as multicast mobile TV, Internet Protocol TV (IPTV) and Video on Demand (VoD) will become dominant applications in the near future. The success of *youtube.com* is another indicator of the above trend. In order to meet the heterogeneity of converged networks and the diversity of user terminals, video delivery must adapt to network changes.

There are two main approaches to this issue in the research community. In the single-rate multicast mode, the sender transmits at one fixed rate, or the rate is adaptive and is defined by either the receiver with the lowest bandwidth capacity [1], or by an inter-receiver fairness objective [2]. In multi-rate multicast schemes a video file is transmitted in a number of different layers (streams). Although single-rate multicasting is easier to implement it has low applicability due to poor scalability.

Multi-rate multicasting exhibits better scalability, is more flexible and can make more efficient use of network resources. Multi-rate multicasting has two basic modes. In the layered mode, each video file is encoded to one base layer and several enhancement layers. The layers may be interrelated for cumulative layered multicast, or may be operated independently. Simulcast is referenced to the transmission of a number of independent streams with the same content that differ in quality and hence in bandwidth requirements. The advantage of having different versions of the same content is that it does not require more sophisticated encoders.

Simulcast technology is fairly simple when compared with layered encoding. The drawback, however, is that the multiple versions of the same multimedia information are transmitted over the network in parallel (something that might be considered as a waste of bandwidth). This is done so that users can choose the appropriate version at any given time. Therefore, it is important to minimize the number of transmitted streams to free up network resources. This can be feasible only if the transmitted streams are adaptive, so that they can serve a large number of users with similar receiving capabilities.

Another major issue which is related to the transmission of multimedia data is the "Transmission Control Protocol (TCP)-friendliness". To explain this issue in more detail, consider that real time video and Voice over IP (VoIP) applications use the User Datagram Protocol (UDP) on top of the Real Time transport Protocol (RTP) [3] to send information packets. UDP however, does not employ any congestion control mechanism whilst TCP does. The impact is that bandwidth allocation is unfairly distributed between applications (TCP-based and multimedia applications). This may cause widespread TCP applications (web browsing, e-mail, ftp e.t.c) to starve when sharing the same network resources with multimedia applications.

All the above open issues motivated the design of the Adaptive Smooth Simulcast Protocol (ASSP), which is a new multi-rate transport protocol for simulcast transmission over best-effort networks. The initial introduction of ASSP with preliminary performance evaluation results is cited in [4]. In this extended version we provide more details on the protocol's architecture and focus on a performance evaluation with both video quality and network-centric metrics in an integrated simulation environment. Furthermore, we compare ASSP against another solution in the area of multi-rate congestion control schemes.

During the design of ASSP we focus on multimedia transmission over the Internet and we design ASSP to serve as the transmission protocol in unmanaged wired networks without any support in the intermediate elements (routers) other than multicast support. The multimedia transmission over control network (for example ISP controlled network) is out of scope of this paper.

The key attributes of ASSP are: (a) TCP-friendly behavior, (b) adaptive per-stream transmission rates, (c) adaptive scalability to large sets of receivers, and finally, (d) smooth transmission rates that are suitable for multimedia applications.

The building block of ASSP is based on a single-rate multicast protocol named Adaptive Smooth Multicast Protocol (ASMP) ([5, 6]). Performance evaluation results [6] show that ASMP is a serious competitor to well-known single rate schemes such as TCP-Friendly Multicast Congestion Control (TFMCC) [1] and the Pragmatic General Multicast Congestion Control (PGMCC) [7].

ASSP is the extension of ASMP from the single-rate multicast schemes to simulcast transmission. As a result, the transmission of each multicast stream, in the context of ASSP, is based on the underlying congestion control mechanism that is implemented by ASMP. ASSP itself is responsible for handling all the issues related to simulcast transmission and the management and synchronization of the multiple multicast streams.

ASSP exploits the concept of "*smooth transmission*" to avoid large oscillations of the transmission rate of each individual stream and to minimize the join and leave attempts to various multicast streams. This is related to long leave latencies of the Group Management Protocol (IGMP) [8]. To explain this consider a host that is changing its subscription level by leaving one session and joining a new session which is the closest session to its bandwidth capabilities. The time to complete the process of leaving a session is termed as "leave latency". The long leave latencies of IGMP preserve network resources (e.g. bandwidth) but cause congestion. If the transport protocol is very reactive to instantaneous network changes it will cause oscillations and increase congestion due to frequent join and leave requests from one session to another.

Another important attribute of ASSP is "*TCP-friendlines*s" as each individual receiver calculates a TCP-friendly bandwidth share with the use of the TCP analytical model.

Lastly, ASSP is a pure end-to-end solution that does not require any network support except for IP-multicast. That means that ASSP can be employed at the end hosts (sender and receiver) leaving the intermediate network elements (routers) untouched. This is an important attribute as it allows easy deployment over different administrative domains. Furthermore, we develop ASSP up to the application layer without affecting the Operating System (OS) and the Internet protocol stack.

We investigate the performance of ASSP by conducting a number of experiments and compare ASSP against the Smooth Multi-rate Congestion Control (SMCC) [9] which is another solution in the area of multi-rate congestion control schemes.

The rest of this paper is organized as follows: In the next section, we discuss recent advances in the area of multi-rate multicast protocols for multimedia transmission. In Sect. 3, we provide a detailed description of ASSP and explain the internal functions of the protocol. We discuss the video quality assessment methods in Sect. 4. The simulation environment for our experiments is discussed in Sect. 5. The performance evaluation of ASSP is presented in Sect. 6. In Sect. 7, we compare ASSP against SMCC. We conclude our paper in Sect. 8.

2 Related Work

The research community has provided a variety of new and promising proposals for multi-rate multicast transmission that are mainly designed for multimedia applications. The first practical adaptation protocol for layered video multicast was the Receiver-driven Layered Multicast (RLM) [10] protocol. RLM uses a predetermined number of fixed transmission rates. Receivers join the multicast group that best satisfy their Quality of Service (QoS) criteria. The sender has no active participation in the congestion control process except for the transmission of the different layers in separate multicast groups. Thus, receivers add an additional layer, which increases the quality of the multimedia data, when they observe spare link capacity. Alternatively, they drop the additional layer when they observe congestion. In Packet pair receiver-driven Layered Multicast (PLM) [11] receivers add and drop layers according to the available bandwidth in the bottleneck link. That was an improvement over RLM and is done to mitigate the effects of falsely adding a higher layer that increases congestion. With the use of packet pairs, receivers normally would not add any layer that leads to higher receiving rates than the capacity of the bottleneck link. The sender in PLM has the same passive role as in RLM, which is restricted to the transmission of the different layers without performing any kind of adaptation. The transmission rates of the various layers (streams) in PLM are fixed and hence they do not follow the network dynamics. Fair Layered Increase/Decrease with Dynamic Layering (FLID-DL) [12] tries to mitigate known drawbacks that are related to long IGMP leave latencies and which leave the network in a congested state. It uses dynamic layers in the sense that the sender changes the transmission rate of each layer over time. Receivers in FLID-DL can decrease their transmission rates by not joining any additional layers. In order for receivers to maintain a given reception rate they must periodically join additional layers at a moderate pace. A comparison of FLID-DL against PLM is presented in [13]. The authors use network-centric metrics and the network simulator (ns-2) [14] for their study. Their performance evaluation criteria are similar with the recently published RFC 5166 [15]. Simulation results show that PLM outperforms FLID-DL in all tested scenarios. A rate adaptive multicast congestion control scheme is presented in [16]. The transmission rate is defined by the receiver with the smallest bandwidth capability. The protocol seems to be TCP-friendly but suffers from high escalations, especially when trying to follow network and traffic dynamics. Fine-grained layered multicast [17] attempts to minimize the drawbacks of cumulative layering regarding that relies on coarse-grained congestion control. It proposes a non-cumulative layer joining approach in which receivers infer the level of congestion in the network by using packet losses, and adjust their subscription level accordingly. STAIR (Simulate TCP's Additive Increase/multiplicative decrease with Rate-based) [18], further minimizes the IGMP control traffic with the concept of *"stair layers"*. The stair layer is the layer whose rate dynamically increases over time from a base rate of one packet per Roundtrip Transmission Time (RTT) up to a maximum rate, before dropping back to the base rate. STAIR simulates the behavior of a TCP unicast connection with the use of an Additive Increase Multiplicative Decrease (AIMD) congestion control algorithm. However, it introduces large oscillations of the transmission rates and has the well-known "*saw tooth*" pattern.

In the simulcast research area a representative proposal is the Destination Set Grouping (DSG) [19]. The source in DSG transmits three streams of the same video content at low, medium and high quality. A receiver subscribes to a stream that is closest to its requirements. DSG uses internal mechanisms to adjust the transmission rate of each stream within a predefined range. In addition to sender-based adaptation, DSG also supports receiver-based adaptation. It allows the receiver to change to a lower or higher capacity stream to better satisfy its requirements. In inter-receiver fair multicast [20] the rate of each multicast stream is adaptive with the goal of maximizing the inter-receiver fairness: an intra-session measure that captures the collective *"satisfaction"* of the session receivers. Sender-Receiver based Adaptation scheme for Multicast Transmission using Simulcast (SRAMT-S) [21] is a newer proposal in which the transmission rate of each multicast stream is based on joint sender-receiver estimations. SRAMT-S offers better scalability as the transmission rate of each multicast stream is dynamically adjusted to network changes.

Actually, simulcast solutions have already been implemented in many commercial video-streaming systems. RealNetworks' RealSystem G2 supports simulcast under the name of SureStream, which generates a fixed number of streams at predefined rates. A receiver can dynamically choose a stream that is closer to its bandwidth capabilities.

3 Adaptive Smooth Simulcast Protocol (ASSP)

3.1 Brief Description of ASSP

The key idea of ASSP is that each stream is not transmitted at a fixed rate. Streams are adaptive so that they can accommodate a large number of users with "similar" bandwidth capabilities. In this way, we can minimize the number of required streams in a simulcast transmission in order to save network resources. The streams, however, need to be bounded between predefined lower and upper limits. The innovation in ASSP lays in the way the receivers make the decision to join or leave a higher or lower quality stream, based on network statistical measurements and a *"strict"* decision-making algorithm. This algorithm minimizes the frequent and false attempts to join a lower or higher quality stream that cause instability and

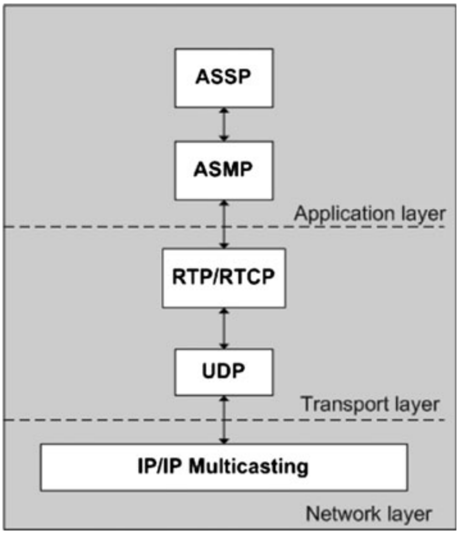

Fig. 1 ASSP protocol stack

congestion. In this way, receivers obtain a certain confidence level before they make such decisions. The RTP Control Protocol (RTP/RTCP) protocol is used for the dissemination of control messages between the sender and the receivers in the various multicast groups. Scalability is ensured by the underlying ASMP protocol, which provides the mechanisms for receiver feedback suppression. Figure 1 depicts the ASSP protocol stack. Mathematical notations of ASSP are described in Table 1.

ASSP can be easily deployed onto unmanaged networks due to the following: (a) both the source and receivers only require simple computations and these computations are independent of the number of receivers (high scalability), (b) we do not make any assumption on intermediate network support other than standard multicast capabilities, and (c) ASSP is a pure end-to-end solution that is implemented up to the application layer leaving the network elements and the operating system untouched. ASSP is also *effective* because, (a) it is TCP-friendly, since the underlying congestion control mechanism at each individual stream is TCP-friendly, (b) the protocol does not disseminate additional control messages other than the RTCP sender and receiver reports, and (c) ASSP performs well in our simulations. A high-level overview of the functionality of ASSP is presented below:

- The receiver measures a smooth TCP-friendly bandwidth share with the use of the TCP analytical model and statistical data that is related to network conditions.
- The receiver compares this TCP-friendly bandwidth share with both the sender's transmission rates in all streams and the limits of the upper and lower streams. In predefined time slots, the receiver can leave and join a lower or higher capacity stream based on a decision-making algorithm (Eq. 7).
- The sender gathers the RTCP receiver reports and performs per-stream transmission rate adaptations based on the reported values.

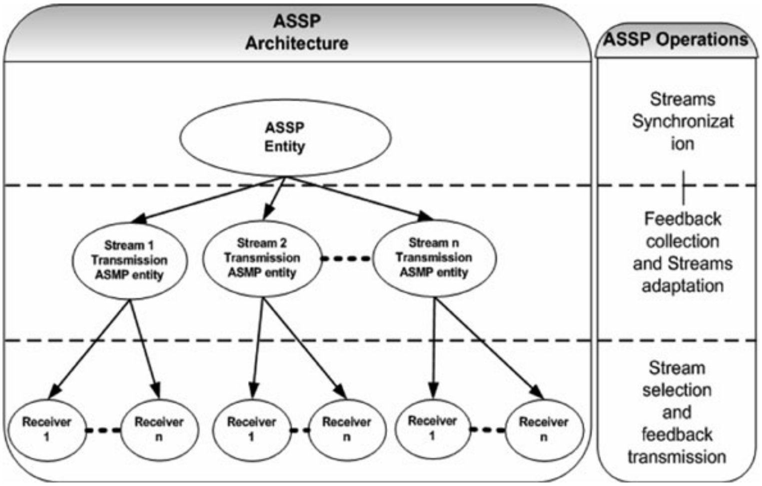

Fig. 2 ASSP architecture

- The sender includes the average transmission rate of each stream in the application part of the RTCP sender reports.
- At fixed time-intervals the sender notifies all receivers, so that join and leave requests are synchronized.

In the following paragraphs, we present all the issues related to simulcast transmission, the management and synchronization of the multiple multicast streams. The transmission of each single multicast stream is controlled by the ASMP protocol (Fig. 2).

3.2 The Underlying Congestion Control Protocol

As we previously explained, we use ASMP as the underlying congestion control mechanism for each individual simulcast stream. ASMP is explicitly designed to operate over wired networks in which packet losses occur mainly due to congestion. The receiver emulates the behavior of a TCP agent and when packet losses occur, it estimates a TCP friendly bandwidth share r^i_{tcp} in every RTCP report interval with the use of the following analytical model presented in [22]:

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

Where, r^i_{tcp} is the estimation of the transmission rate of receiver i (in bytes/sec), P is packet size in bytes, l is the packet loss rate, t_{out} is the TCP retransmission timeout, t_{RTT} is the RTT time of the TCP connection and D is the number of acknowledged TCP packets by each acknowledgment. In our implementation 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). In order to avoid abrupt changes of the transmission rate we define that when the receiver has not experienced any packet losses since the previous RTCP report, the r^i_{tcp} must not be increased by more than one P/RTT (in bytes/second). For this reason we define the following equation for the receiver calculation about the new r^i_{tcp} value (in bytes/sec):

$$r^i_{\mathrm{tcp}} \leftarrow r^i_{\mathrm{tcp}} + \frac{1}{t_{\mathrm{RTT}}} P \tag{2}$$

More information about ASMP can be found in [5] and [6].

3.3 Sender's Feedback Functions

Sender's feedback functions are implemented via ASMP entities (we also call them stream managers) that gather the RTCP receiver reports and exploit the received feedback reports of each stream (Fig. 3). The sender performs the following procedures in the event of a newly arrived RTCP receiver report:

$$\begin{aligned} &\textbf{Receive RTCP packet}: \\ &compare() \\ &adjustTransmissionRate() \end{aligned} \tag{3}$$

$$\begin{aligned} &\textbf{Subroutine Compare}: \\ &\textbf{for}\ all\ receivers\ in\ stream\ j \\ &tx\_inst_j(t) \leftarrow \min\left(rx\_inst_j^1 \ldots rx\_inst_j^i\right) \end{aligned} \tag{4}$$

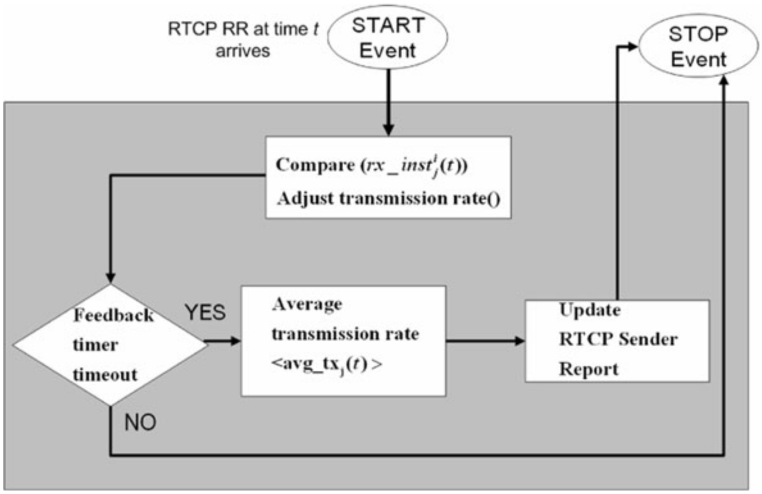

Fig. 3 Feedback handler at ASSP sender

In other words, the stream manager of stream j compares the reported $rx\_inst_j^i(t)$ at time t from receiver i with all previous reported values from all receivers that belong to stream j. The new transmission rate (set by the subroutine *adjustTransmission-Rate()*) of stream j is the lowest reported $rx\_inst_j^i(t)$ value. Then the sender adds the $tx\_inst_j(t)$ value in a list and in the event of feedback timer-timeout it averages the transmission rate of steam j as follows:

$$avg\_tx_j(t) = \frac{1}{\Delta t} \int_{t}^{t+\Delta t} tx\_inst_j(t)\, dt \tag{5}$$

The sender includes the average transmission rate of each stream in the application part of the sender RTCP report. Therefore, at any given time, each receiver has knowledge of the average transmission rate of each multicast stream. In this work, we set the timeout interval for averaging the transmission rates to 5 s based on simulation results, which show that the above value is a good compromise between responsiveness and subscription level accuracy. Lower timeout values do not provide the desired confidence level for such decisions. Therefore, for implementing ASSP we suggest a timeout value of 5 s, which provides high statistical accuracy without affecting the responsiveness of the protocol to network dynamics.

3.4 Receiver's Feedback Functions

The ASSP receiver is responsible for monitoring the reported transmission rates from the sender and adjusts its subscription level[1] based on a decision-making algorithm. Upon the arrival of a new RTCP packet, the receiver checks the join flag (Fig. 4). We use the application part of the RTCP packet to set a flag in order to provide receivers with a notification concerning the join/leave requests to a higher or lower capacity stream. When the flag is true, receivers measure the average receiving rate over a period Δt. However, when a receiver leaves its current stream and joins a higher or lower stream there is a period in which it does not receive any data packets. If the measurements for the receiving rate were based only on instantaneous values, the receiver would appear to have zero receiving rates during this leave period. This situation will lead to oscillations when receivers change their subscription level. Therefore, we use an Exponentially Weighted Moving Average (EWMA) function $\Phi(\,)$, with averaging factor $0 < \alpha < 1$.

$$\begin{aligned} &\textbf{Receive RTCP packet}: \\ &\mathit{if}\ (\mathit{flag} = \mathit{true})\ \mathit{then} \\ &\mathit{compare}(avg\_rx^i(t)) \\ &\mathit{else\ do\ nothing} \\ &\mathit{end\ if} \end{aligned} \tag{6}$$

With the use of the EWMA function, we minimize the IGMP leave latency side effects. The following operations take place when a RTCP packet is received at

[1] The subscription level is the stream in which the receiver makes a join request.

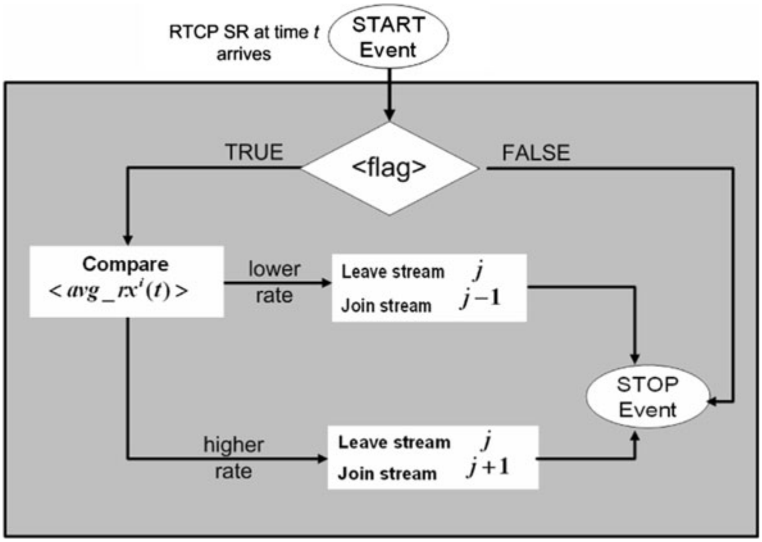

Fig. 4 Feedback handler at ASSP receiver

receiver i. During the connection to the service the sender informs each receiver about the number of simulcast streams and the thresholds of each stream. This feature can be easily implemented with the use of application defined (APP) RTCP packets:

$$
\begin{aligned}
&\textbf{Subroutine Compare}: \\
&if\,(avg\_rx^i(t) > avg\_tx_{j+1}(t) \cdot \beta \,\textbf{and}\, avg\_rx^i(t) > threshold_{J+1} \cdot \lambda) \\
&\quad leave\ stream\ j \\
&\quad join\ stream\ (j+1) \\
&else\ if\,(avg\_rx^i(t) < threshold_j \cdot \kappa) \\
&\quad leave\ stream\ j \\
&\quad join\ stream\ (j-1)
\end{aligned}
\tag{7}
$$

The average receiving rate $avg\_rx^i(t)$ at time t is defined as follows:

$$
\begin{aligned}
avg\_rx^i(t) &= \Phi(avg\_rx^i(t_0), \alpha) \\
avg\_rx^i &= (1-a) \cdot avg\_rx^i(t_0) + a \cdot avg\_rx^i(t_1)
\end{aligned}
\tag{8}
$$

The different values of a define the level of a receiver's responsiveness to network changes. High values make the receiver respond faster but they create unnecessary leave and join requests. Our objective is however, to stabilize the receiver's behavior, avoid frequent join and leave requests. Therefore, we do not let instantaneous high or low $avg^i_{rx}(t)$ values play a central role in the decision-making algorithm. However, as we can see from algorithm (7) the decision to leave a lower capacity stream and join a higher capacity stream is not only based on measuring the average receiving rate. The receiver is not only required to have an average receiving rate that is higher than the lower limit of the higher stream, but it has to be able to follow "similar" receiving rates to those of the set of receivers in this higher stream. In our simulations we set $a = 0.3$, $\beta = 0.7$, $\kappa = 0.8$ and $\lambda = 1.2$ based on various experimentations with different network topologies which are not presented in this paper due to space limitations. Generally, β takes the following values:

$$0.7 \leq \beta \leq 1 \tag{9}$$

which means that the receiver should be able to achieve receiving rates at least 70% of the average transmission rate $tx\_inst_{j+1}(t)$ in order to join the higher capacity stream $j + 1$. The higher stream join factor λ is bounded between 1 and 2

$$1 \leq \lambda \leq 2 \tag{10}$$

meaning that the average receiving rate $avg\_rx^i(t)$ should be at least equal to or higher than the lower limit of the higher capacity stream. Lastly, κ is bounded between $0.8 \leq \kappa \leq 1$. Receivers can remain in the current stream j even though its average receiving rates $avg\_rx^i(t)$ are at least 80% of the lower limit of the current stream. The meaning of the above filtering functions is that receivers build a certain level of confidence when they decide to join a higher capacity stream. On the other hand, receivers can still participate in the current session of stream j even though their receiving capacities are within an acceptable value that is lower than the low limit of the current stream. Therefore, we prevent oscillatory behavior and ensure stability. However, the penalty we pay is lower bandwidth utilization as receivers do not rapidly join or leave a higher or lower capacity stream based on current network conditions. We argue, however, that in many cases stable reception rates close to optimal values are more important than highly oscillatory behavior due to frequent stream changes.

3.5 Extensions to RTP/RTCP Protocols

ASSP functionality is based on the information exchange between the sender and receivers which is implemented by the RTP/RTCP protocol. RTP/RTCP provides an extension mechanism to allow individual implementations that require additional information to be carried in the RTCP packet header. ASSP uses this extension mechanism in order to include the following fields in the RTCP packet header:

- *J_flag*: A *boolean* value to provide receivers with a notification concerning the join/leave requests to a higher or lower capacity stream.
- avg_tx_j, $j = 1, \ldots, n$: The average transmission rate of each simulcast stream.
- $rx\_inst_j^i$, $j = 1, \ldots, n$, $\quad i = 1, \ldots, n$. The instantaneous receiving rate of receiver i of stream j.

The implementation of the above extensions is simple and straightforward. It is also very important that we do not need to disseminate additional control messages other than the RTCP sender and receiver reports.

4 Video Quality Assessment Methods

In this paragraph we discuss video quality assessment methods and metrics that are used for the evaluation of ASSP. What we are interested in is to measure the quality of the perceived video file by the end-user. This is termed as Quality of Experience (QoE) and defined by the ITU-T in [23] as "*the overall acceptability of an application or service, as perceived subjectively by the end-user. It includes the complete end-to-end system effects (client, terminal, network, services infrastructure, etc.) and may be influenced by user expectations and context*". The interesting and difficult part in this process is how to choose the methods and the metrics for assessing the QoE.

There are broadly two categories of methods for assessing the perceived video quality according to the involvement of human interaction during the evaluation process. In the subjective test methods the perceived video quality is defined through human grading in which the individual viewer determines the quality level. Subjective video quality assessment methods are defined by ITU-T in [24].

Objective test methods do not involve human interaction and are classified into three categories. In the first category, the evaluation of a transmitted video is performed by comparing the complete decoded video sequence at the end user to the original one sent by the sender. In the second category, we compare only a part of the features/metrics of the original with the decoded video and not the whole video sequence. In the third category we do not conduct any comparison between the original and the decoded video at the end user, but assess only the decoded video at the end user. The Video Quality Expert Group (VQEG) names these methods as the full, the reduced and the no reference methods [25].

QoE requirements for video and audio may be based on subjective evaluation metrics [26] such as the Mean Opinion Score (MOS) in which a number of viewers determine the video quality in a range 1–5, where 1 is the lowest perceived video quality and 5 the highest quality (Table 2). Although MOS is an effective way to

Table 1 Mathematical notations

Symbol	Meaning
$rx\_inst_j^i(t)$	Instantaneous TCP-Friendly bandwidth share of receiver i in stream j at time t
$tx\_inst_j(t)$	Instantaneous transmission rate of stream j at time t
$avg\_tx_j(t)$	Average transmission rate of stream j at time t
$avg\_rx^i(t)$	Average receiving rate of receiver i at time t
$\Phi()$	EWMA averaging function
α	Exponential averaging factor
β	Transmission rate factor
κ	Current stream leave factor
λ	Higher stream join factor
Δt	Time period over which join or leave decisions are made
j	Stream j
$threshold_j$	Low BW limit of stream j

Table 2 ITU-R Quality and impaired scale [28] and possible PSNR to MOS mapping [29]

PSNR (dB)	MOS	Perceived quality	Impairment
>37	5	Excellent	Imperceptible
31–37	4	Good	Perceptible, but not annoying
25–30	3	Fair	Slightly annoying
20–24	2	Poor	Annoying
<20	1	Bad	Very annoying

measure the QoE of any multimedia service for a user, is considered as time consuming and requires a large number of users to provide reliable results.

To overcome the above limitations in our work, we use the objective full reference test method and calculate the Peak Signal to Noise Ratio (PSNR) [27] by directly comparing the original video file sent by the sender with the decoded video at the end user on a frame-by-frame basis. Then the PSNR values of all individual video frames are averaged to produce the mean PSNR of the complete video sequence. This is then mapped to the corresponding MOS value (Table 2). We need to point out, however, that PSNR mapping to MOS values provides only a rough estimation of the perceived video quality by the end user. For more accurate results real experiments with a sufficient number of viewers should be conducted.

5 Simulation Environment

Most of the related work has been evaluated through simulations conducted with the ns-2 simulator software. Those simulations were not based on any multimedia traffic generation model and in the best case, trace files were used instead. Therefore, the only quality indicators used in these simulations were purely based on "classic" network metrics (throughput, delay, packet losses, etc.). However, different multimedia encodings can result in different perceived video quality, although the transmission is done with exactly the same set of protocols and under the same network conditions. Therefore, it is important to study the performance of any proposed solution for multimedia data transmission by using real video files and to associate simulation results with video QoS metrics. For the purpose of this work, we have extended the Evalvid-RA [30] tool-set in order to conduct a number of realistic experiments with real video files.

This simulation environment consists of three parts and is depicted in Fig. 5. During the pre-processing phase, a raw video file, which is usually stored in YUV format, is encoded with the desired video encoder into 30 different encoded MPEG-4 video clips with quantizer scale values in the range 2–31. Quantizer scale 2 provides an encoded video with the highest quality. We use the *ffmpeg* [31] free video encoder for the creation of the video clips. For our simulations, all video clips have temporal resolution of 25 frames per second and GoP (Group of Pictures) pattern IBPBPBPBPBPB, with a size of 12 frames. The frame size of all clips is 352 × 288 pixels, which is known as the Common Intermediate Format (CIF). After all the video files are encoded they are then traced to produce 30 frame-size

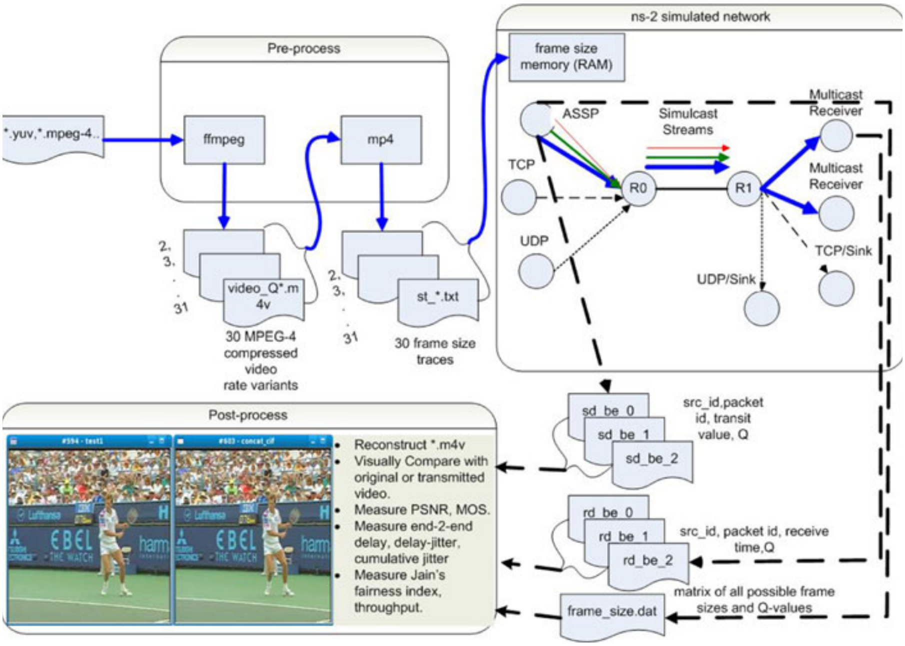

Fig. 5 Overview of the simulation environment (pre-processing, ns-2, post-processing)

traces files. At the end of the pre-processing phase, we thus have 30 *m4v* files with their associated frame size files.

Next, the ns-2 creates a simulated network. The video file is transmitted from the server by using a number of predefined simulcast streams to different groups of receivers. In our simulations, we use high, medium and low quality streams. During the simulation run-time we store the traces of all transmitted simulcast streams from the sender and the traces from all the multicast receivers.

The third part of the simulation environment consists of the reconstruction of the transmitted video and the measurement of the performance evaluation metrics. The reconstruction of the received video traces is implemented off-line by comparing the transmitted and the received traces with those of the original video sequence of all the transmitted simulcast streams. Next PSNR is calculated by comparing the original video file with the received file at the end user on a frame-by-frame basis. The following metrics are also stored and calculated:

- PSNR/MOS values
- End-to-end delay
- Packet delay variations
- Inter-frame cumulative jitter
- Delay jitter
- Frame loss rate
- Throughput per flow
- Jain's fairness index

6 Performance Evaluation

In this section we present simulation results that are used for the evaluation of ASSP. Our evaluation is based on both network-centric metrics in accordance with the recent RFC 5166, and QoE requirements for video (MOS) based on the ITU-T Recommendation G.1080. Therefore we conduct simulations in order to investigate the following:

- The ability of ASSP to detect the available bandwidth at the link level and adjust the transmission rate per simulcast stream.
- Delay and cumulative jitter measurements which are connected with the required playback buffer size of a video transmission.
- The fairness of ASSP towards competing TCP traffic.
- The fairness among flows of the same protocol.
- The stability of ASSP under changing network conditions, especially its ability to prevent oscillations (smoothness).
- The ability of ASSP to adapt network changes due to presence of any competing traffic
- The QoE performance based on MOS grading.

6.1 Level Subscription Accuracy

In this simulation, we investigate the accuracy of ASSP in terms of stream level subscription and its stability. The simulation scenario consists of one multicast sender (S) and six multicast receivers, ASSP1 to ASSP6 (Fig. 6). C1 to C4 are four network routers. We set up Drop Tail queues in the routers C1 to C4 and set the one-way delay in all paths to 50 ms. We use a medium-motion video sequence ("highway") which consists of 2000 frames. This medium-motion sequence is challenging as it provides rather low PSNR values. The sender transmits three multicast streams each with a different quality. To do this, at start time the server initiates the transmission of the different streams with different quantizer scales. However, during the simulation, the sender adjusts the transmission rate of each individual stream based on feedback reports from the receivers. As for the ASSP multicast receivers, they are connected with links that differ in capacity. Therefore, we create three different groups of receivers and each group has different receiving capabilities. At start time, all receivers join the stream with the lowest quality. For easier observation, in our simulation graphs we plot the results of only one receiver out of each multicast pair. We observe that at the first join attempt, ASSP3 and ASSP5 join the medium capacity stream and ASSP5 then quickly joins the high capacity stream, (Fig. 7). It is interesting to observe that if there are no changes to the network conditions, the receivers present stable behavior throughout the simulation lifetime and do not "jump" from lower to higher streams and vice versa.

The average throughput of each group is 83, 221 and 360 Kb/s, with a frame loss ratio of 1.6, 1 and 2% for the low, medium and high quality stream, respectively. We notice this rather low link utilization, which is inherited from the smoothing property of the underlying congestion control mechanism. Our design objective is to

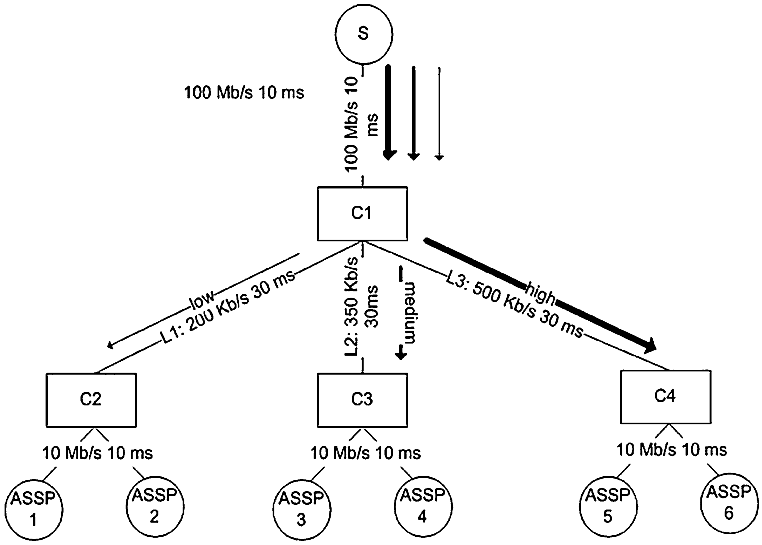

Fig. 6 Topology for stream accuracy subscription

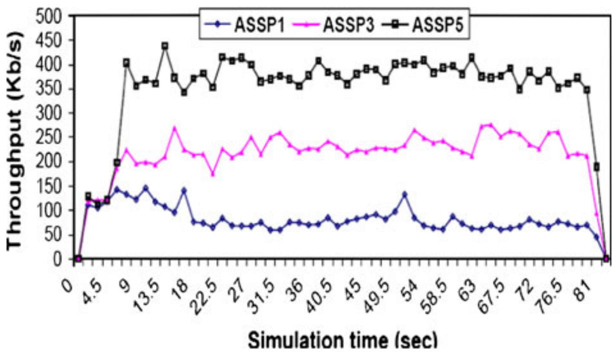

Fig. 7 Achieved throughput of the three multicast receivers

minimize oscillations, as they tend to increase packet losses and lead to an unsatisfactory video experience at the user side. However, there is always a trade-off between maximizing throughput and stability.

We measure the stability of ASSP by using the coefficients of variation (CoV)[2] of the throughput values and plot the results in Fig. 8. We observe that ASSP shows good stability. Delay jitter measurements (Fig. 9) present low values which are suitable for conversational applications as the delay jitter is much lower than 150 ms. Cumulative jitter [32] (Fig. 10) presents high values in the case of the low

[2] Coefficient of Variation (CoV) is the standard deviation divided by the mean.

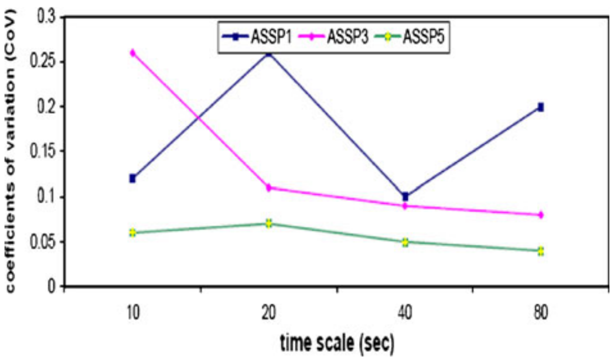

Fig. 8 CoV of the three ASSP flows

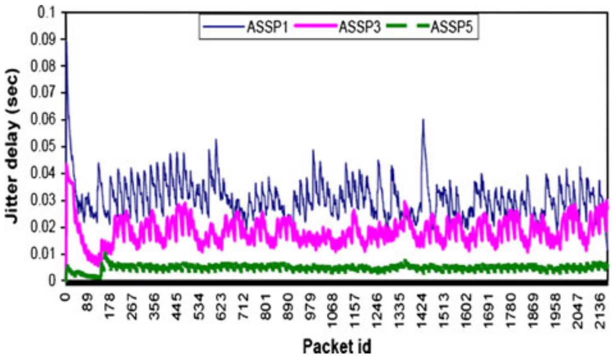

Fig. 9 Delay jitter of the three ASSP flows

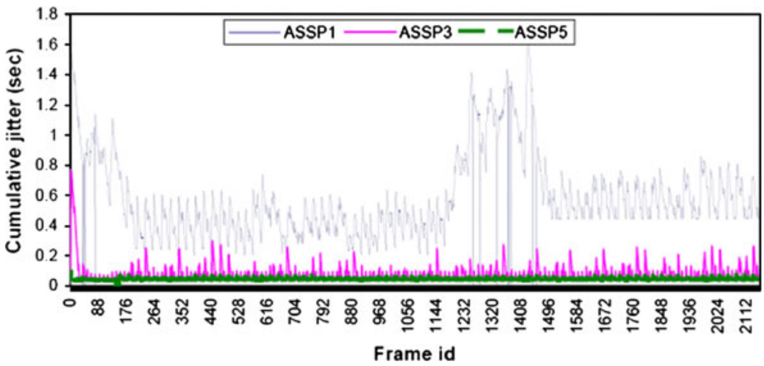

Fig. 10 Cumulative jitter of ASSP flows

quality stream, which is an indication of a congested link between the sender and the low capacity receivers. These values require a higher playback buffer to smooth the negative effects of late-arriving frames.

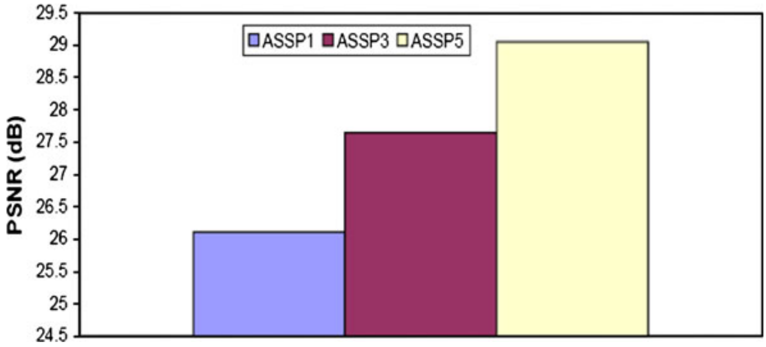

Fig. 11 PSNR values of the received MPEG-4 video

The average PSNR values of the three received video files are depicted in Fig. 11. We observe that on average all the received video files have PSNR values above 25 dB, although in this simulation we use low capacity links that do not allow for the transmission of high quality video files. The best quality video with higher PSNR values could have been obtained by using links with capacities of at least 3 Mb/s. However, in this simulation we intentionally test ASSP using low capacity links to investigate its limitations. The average Mean Opinion Score (MOS) grading for the three flows is 3.0.

Our assessment from this experiment is that ASSP presents stable behavior, without oscillations. PSNR values are within acceptable rates and provide a rather good quality video. The low frame-loss ratio is an indication of the correctness of the underlying congestion control mechanism.

6.2 Fairness

We investigate the fairness of ASSP in two different ways: (a) TCP-friendliness, and (b) fairness between flows of the same protocol. We define a flow as TCP-friendly when it does not consume more bandwidth than a TCP connection which is traversing the same path with this flow. Fairness measurements can include max–min fairness, proportional fairness, Jain's fairness index, and the product measure, which is a variant of network power. In this work, we use the Jain's fairness index which is defined in [33].

6.2.1 TCP Fairness

We simulate the transmission of a video file to a number of multicast groups in an environment that has multiple bottleneck links (Fig. 12). In this topology, the bottleneck links are shared between ASSP and TCP flows. We set up three multicast groups with forty-five ASSP receivers per group, with these groups positioned behind bottleneck links that differ in capacity and delay. We use Drop Tail queues in the routers and set the same packet size for TCP and ASSP in order to obtain a

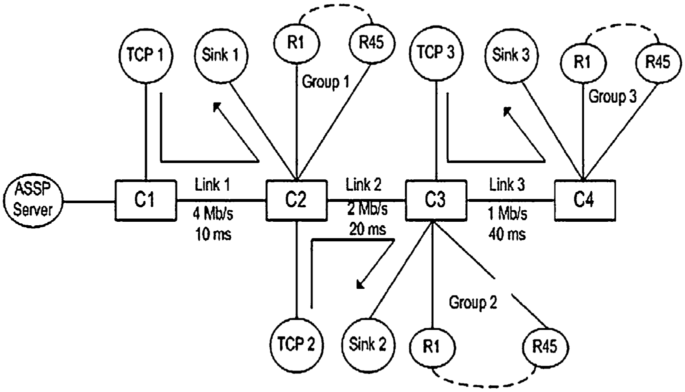

Fig. 12 Network topology for TCP-fairness

fair comparison. At start time, the sender starts the transmission of three multicast streams, each with different quality levels, and all multicast ASSP receivers join the low quality stream. The subscription to a higher or lower multicast stream is based on the functionality of ASSP. Figure 13 depicts the throughput of the background TCP flows and a representative receiver from each multicast group (high, medium and low quality streams). The measured Jain's fairness index is 0.962 indicating that all flows almost equally share the bandwidth of the bottleneck links. The frame loss ratio of the low, medium and high quality streams is measured to be 1.3, 1.6 and 9%, respectively. An initial visual observation is used to observe the steady rates of both the ASSP and TCP flows. By plotting the CoV of all flows, we observe that after the 20th simulation second all flows are in equilibrium with steady rates that do not change for the rest of the simulation lifetime (Fig. 14).

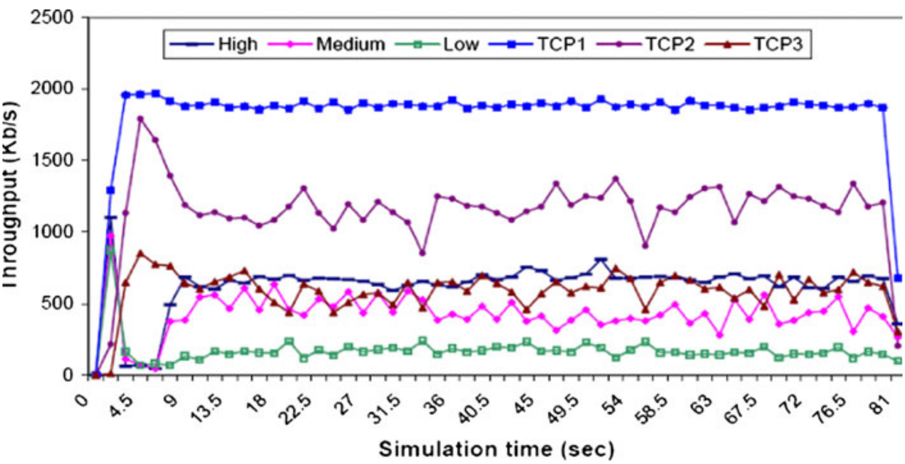

Fig. 13 Throughput plot

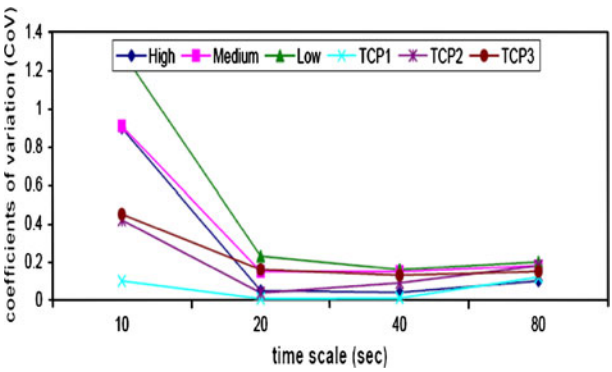

Fig. 14 CoV of all flows (TCP and ASSP)

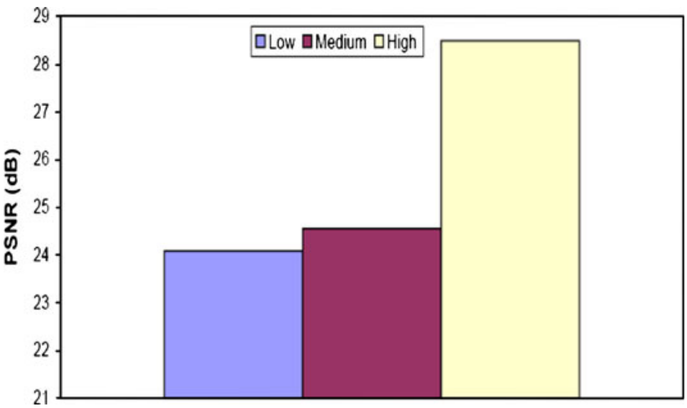

Fig. 15 PSNR values of the received MPEG-4 video

PSNR measurements (Fig. 15) show that all the received video files have on average a fair quality even for ASSP receivers that are utilizing a highly congested low capacity link. At this point, we need to point out that we have succeeded in transferring a video file to different multicast groups with a minimum number of streams, because the underlying congestion control protocol is rate adaptive. If we locate an ASSP agent at the edge of the network, we would probably be able to serve a large number of users that are located behind the same bottleneck links with a very small number of multicast streams. Our assessment from this simulation is that ASSP is indeed a TCP-friendly protocol as it shares network resources with TCP in a fair manner. The smoothing functions of ASSP make it less aggressive in networks with competing TCP traffic. This is an important attribute of ASSP as most of today's user applications in the global web are TCP-based.

6.2.2 Intra-Protocol Fairness

In this simulation, we use the same network topology in Fig. 12 and connect two ASSP sources to C1. There are no competing sources other than the two ASSP sources. In each bottleneck link, we connect two groups of forty-five receivers and each group receives video files from one of the two ASSP sources. Figures 16 and 17 depict the achieved throughput of a representative receiver from the high, medium and low capacity multicast groups. We observe that the results are identical for each of the two competing ASSP sources. The Jain's fairness index in the three bottleneck links is 0.99, showing that network resources are equally shared by the two ASSP sources. The frame loss ratio for the low, medium and high quality streams was measured to be 0.45, 0.05 and 0.15%, respectively. The multicast receivers lost only a few frames due to buffer overflow in the routers.

The CoV (Fig. 18) are very low in all streams and are bounded between 0.2 and 0.005. This is a clear indication of the stability of ASSP. PSNR values (Fig. 19) are higher when compared with the simulation results which involved competition with TCP traffic. This is also a result of very low frame loss ratio. The above simulation results indicate a high level of fairness when ASSP sources compete for network resources. Video quality was also measured to be very high when taking into account the low capacity of the three bottleneck links.

6.3 Performance with ON–OFF UDP Flows as Background Traffic

In this simulation scenario, we replace the TCP sources in the network topology of Fig. 12, with UDP flows. Figure 20 presents results with background traffic provided by ON/OFF UDP flows. For clarity of presentation, we plot the results of one ASSP receiver from each multicast group. The duration of ON and OFF time is set to 10 s in such way that at any given time there is at least one active UDP source in ON mode and thus sending at 500 Kb/s. By plotting the CoV (Fig. 21), we

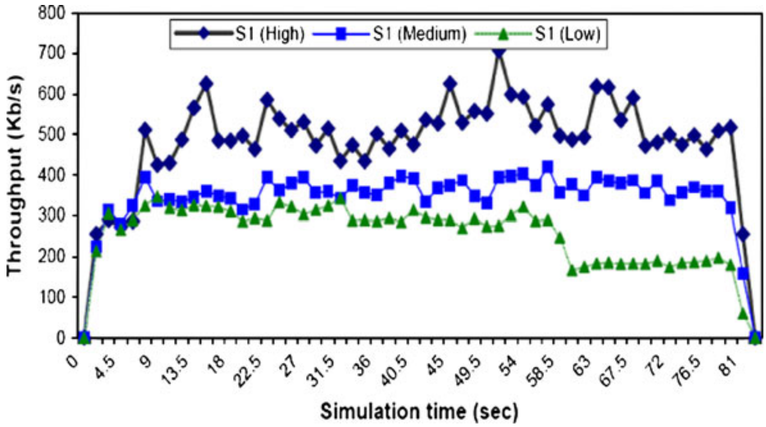

Fig. 16 Achieved throughput (first ASSP source)

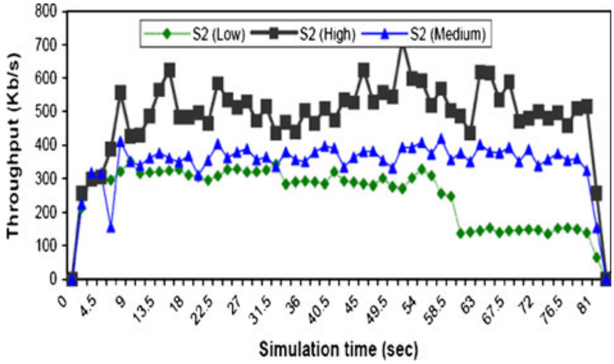

Fig. 17 Achieved throughput (second ASSP source)

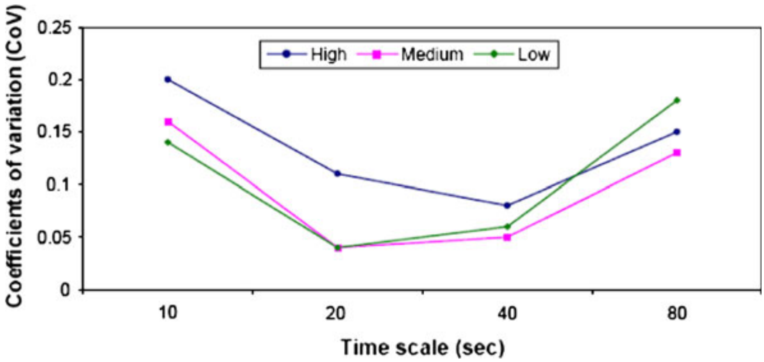

Fig. 18 CoV of three representative receivers

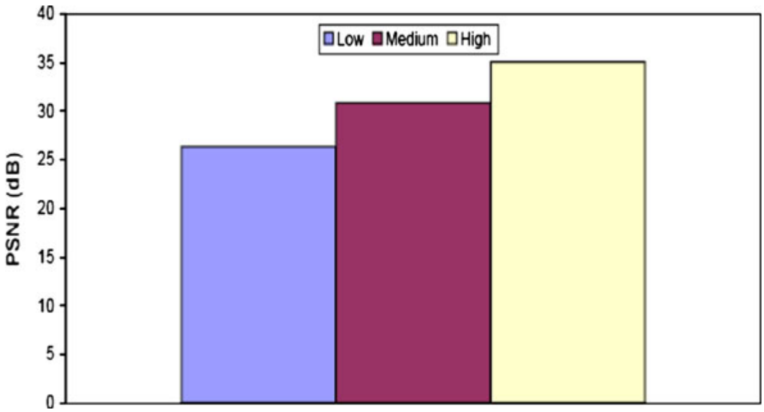

Fig. 19 PSNR values of the received MPEG-4 video

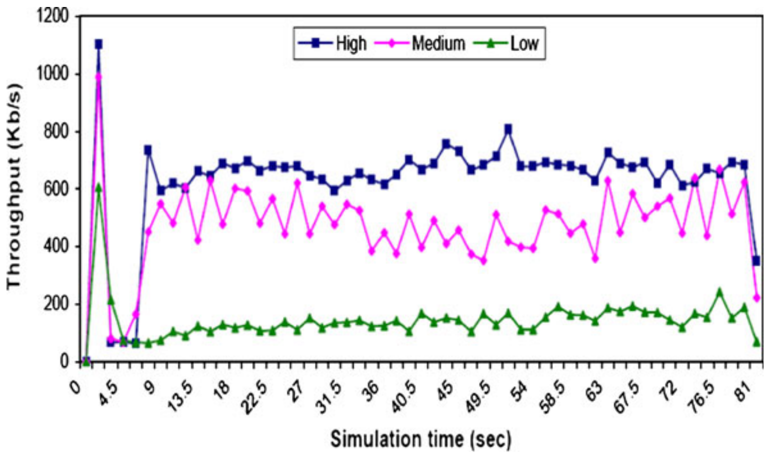

Fig. 20 Achieved throughput of ASSP flows

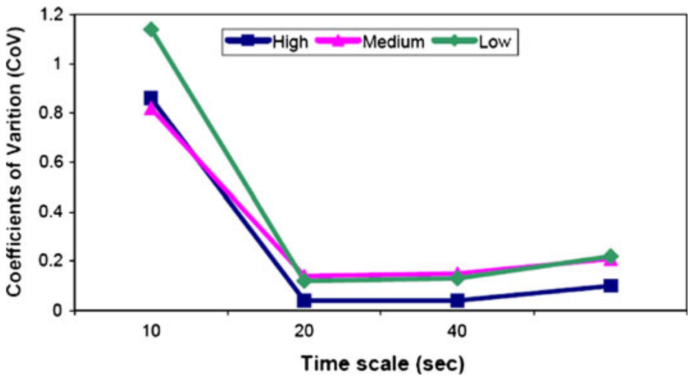

Fig. 21 CoV of all ASSP flows

observe that in the beginning of the simulation the ASSP receivers present a slightly variable behavior. After the 20th simulation second, ASSP presents a stable behavior without any rate variations for the rest of the simulation lifetime. This is an important attribute for video transmission applications.

The frame loss ratio of the three flows (high, medium and low) is 0.7, 0.6, and 2%, respectively, which is rather low for this simulation environment with competing UDP flows. Cumulative jitter measurements (Fig. 22) indicate low values of cumulative jitter for the high and medium quality streams and higher values (above 2 s) for the low quality stream. This is an indication of mild network congestion due to the presence of uncontrolled UDP traffic. We have also observed that the frame loss ratio measurements were higher than those from the previous simulations were. However, our general assessment is that ASSP retains its functionality when competing for network resources with uncontrolled UDP traffic as indicated by the measured RSNR values (Fig. 23).

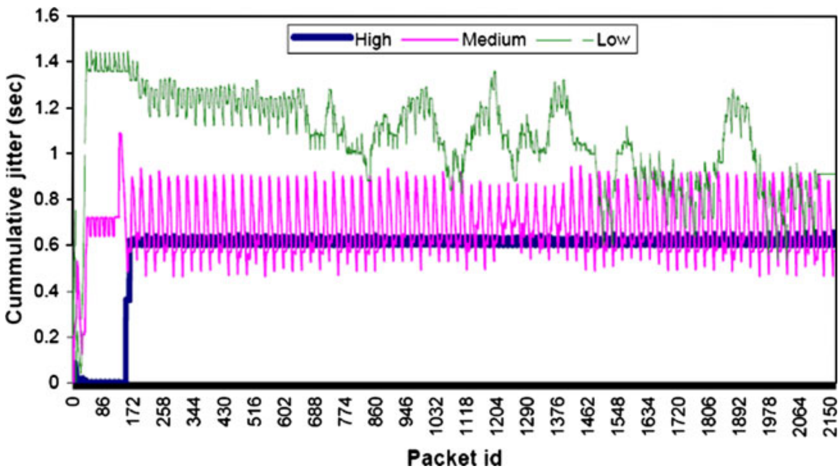

Fig. 22 Cumulative jitter of the three ASSP flows

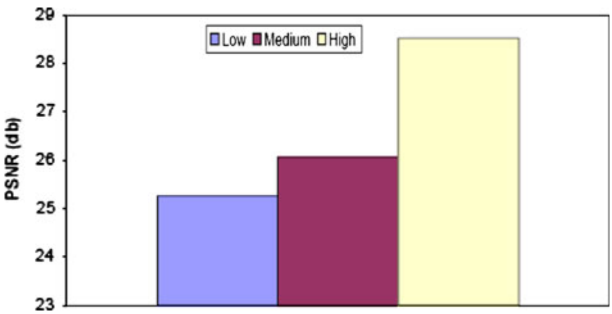

Fig. 23 PSNR values of the received MPEG-4 video

7 Comparison With SMCC

In this section, we present a comparison of ASSP against Smooth Multirate Multicast Congestion Control (SMCC). Actually, we attempt to compare two different schemes in the area of multi-rate multicasting. SMCC is a proposed solution for cumulative layered multicast whereas ASSP is a pure simulcast protocol that generates replicated streams at different rates. SMCC is better suited to the latest advances in video coding (SVC) [34]. Simulcast and layered multicast have been compared in many different contexts, including IP-layer multicasting [35], and TCP-friendliness [36]. Recent studies [37] on IPTV networks indicate that in some cases, layered multicasting is less efficient than simulcast, if for most of the channels one resolution is needed. On the other hand, replication of the same content introduces noticeable redundancy. Therefore, balancing bandwidth consumption with user satisfaction becomes a critical concern in simulcast. With the following experiments we try to provide a deeper investigation into some of the limitations as well as the pros and cons of ASSP and SMCC.

For the purpose of this work, we integrate the publicly available codes of SMCC [38] and ASSP [39] into ns-2 and conduct our experiments under a controlled environment. Performance evaluation criteria of transport protocols are discussed in RFC 5166. Therefore, by taking into account RFC 5166 we investigate the performance of SMCC and ASSP into the following areas:

- The network utilization level and the fairness of each protocol towards TCP traffic.
- The fairness among flows of the same protocol.
- The delay and the packet loss ratio that are introduced by the two protocols.
- The stability of each protocol under changing conditions, especially their ability to prevent oscillations (smoothness).

For our experiments, we use the same network topologies, with each network having identical attributes, to obtain a complete and fair comparison between SMCC and ASSP. The codes, simulation scripts and results are publicly available in [39]. Before the comparison, we provide a brief description of SMCC.

7.1 Smooth Multirate Multicast Congestion Control (SMCC)

SMCC is a multiple-rate equation-based congestion control algorithm for cumulative layered multicast sessions that employs TFMCC as the primary underlying control mechanism for each layer. Since each layer is controlled independently by TFMCC, the properties of TFMCC hold for all participants in any given layer. As such, the layer rates are both dynamic and adaptive. In SMCC, each receiver evaluates, on a per stream basis, a control equation (Eq. 11) that is derived from the model of TCP's long-term throughput [40], and then uses this to directly control the sender's transmission rate.

$$T_{\mathrm{tcp}} = \frac{s}{\mathrm{RTT}\left(\sqrt{\frac{2p}{3}} + \left(12\sqrt{\frac{2p}{8}}\right)p(1+32p^2)\right)} \tag{11}$$

where T_{tcp} is a function of the steady-state loss event rate p, the TCP round-trip time RTT, and the packet size s.

SMCC combines the benefits of TFMCC (smooth rate control, equation-based TCP friendliness) with the scalability and flexibility of multiple rates to provide a multiple rate multicast congestion control policy. In SMCC, the receivers cumulatively subscribe to appropriate layers based on their estimated reception rate using the TCP throughput equation also employed in TFMCC. In addition to the TFMCC functionality, SMCC provides a new additive increase join attempt to avoid abrupt rate increases when the receiver attempts to add an additional layer. However, the calculated throughput using equation-based methods may not provide a sufficiently accurate indication to decide when to join the next layer. To avoid these problems, the receiver in SMCC joins the next layer through the join attempt when its calculated throughput is in the range of the next layer rate. The smooth rate change of SMCC is ideally suited to streaming multimedia applications. It is worth mentioning that SMCC requires no additional router support beyond basic multicast functionality, and does not place any new demands on any existing multicast protocols.

7.2 Network Utilization and TCP-Friendliness

To investigate the ability of each protocol to exploit network resources and their TCP-friendliness we use the simulation scenario in Fig. 24. In this topology, we set up three bottleneck links, each one with a different capacity. Therefore, receivers are connected with links that differ in capacity. Each bottleneck link is shared by two multicast receivers and one TCP flow. C0 to C4 are the five network routers. We set up Drop Tail queues in the routers and set the one-way delay in all paths to 32 ms. In addition, we set the packet size for all flows to 1000 bytes to obtain a fair comparison.

We run the following simulations for ASSP and SMCC. In the ASSP case, the sender transmits three multicast streams with the following limits: Stream1 (300 Kb/s–2 Mb/s), Stream2 (2 Mb/s–5 Mb/s), and Stream3 (5 Mb/s–10 Mb/s). In the beginning of the simulation, all ASSP receivers join the lowest capacity stream, Stream1. For the SMCC simulations, we set the transmission rate of the base layer to 1 Mb/s. We can see (Figs. 25, 26) how the two protocols adjust their transmission rates to satisfy the receiving requirements in the different bottleneck links. The transmission rates of both protocols are dynamically adjusted in respect of the observed congestion in the three different links. With a simple visual comparison,

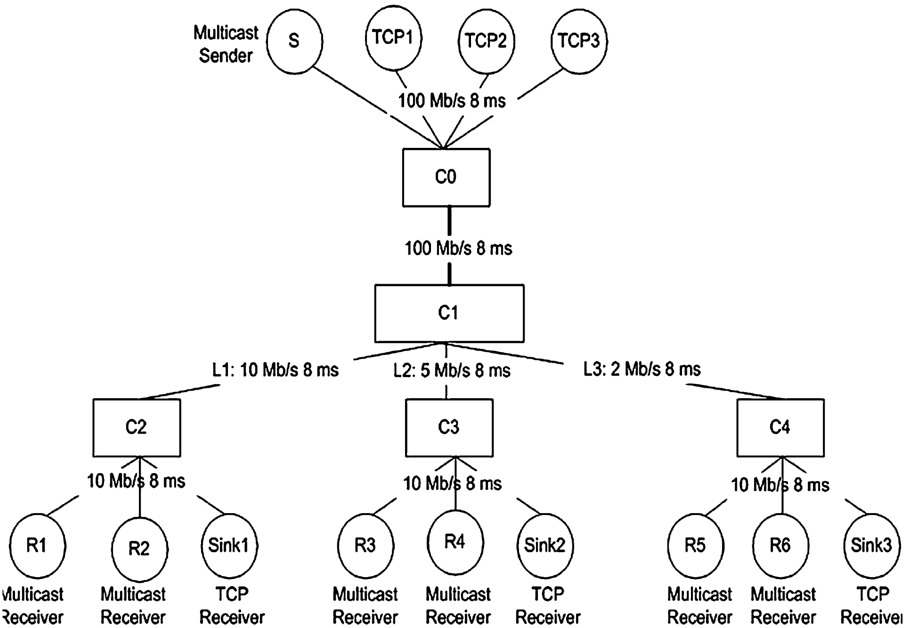

Fig. 24 Bottleneck scenario

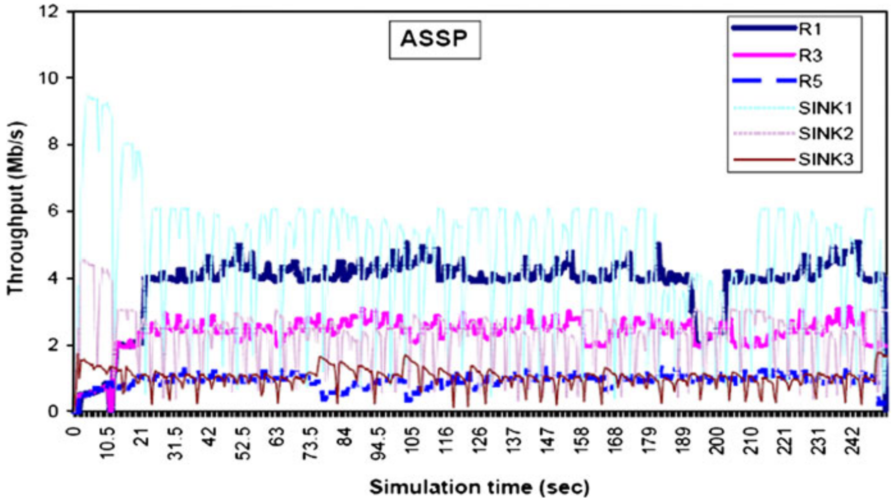

Fig. 25 Transmission rates of ASSP flows

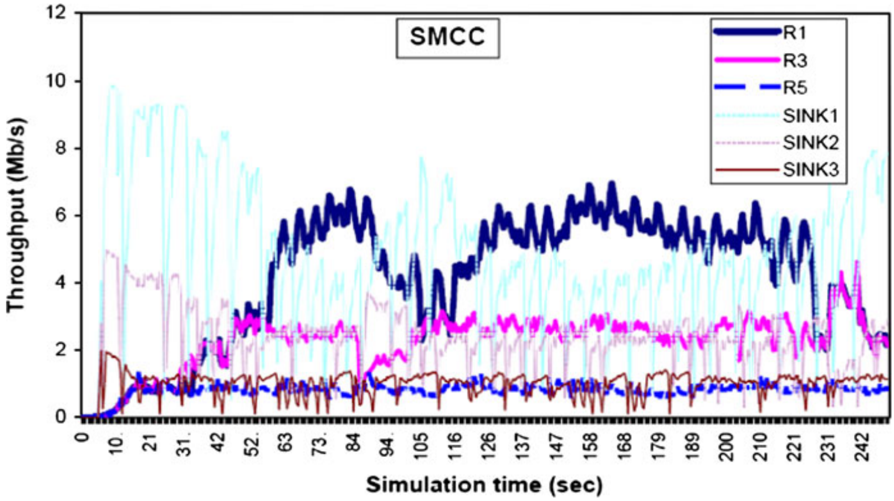

Fig. 26 Transmission rates of SMCC flows

we observe that ASSP presents more stable behavior than SMCC. This however, will be discussed in the next sections in which we investigate the stability of the two protocols in paths with varying RTT values. In Table 3 the simulation statistics of the different flows are given in terms of the achieved throughput. In a "perfect world", network resources should be equally shared by multicast and TCP flows and network utilization would reach up to one hundred per cent. In our simulation, ASSP presents lower performance in the higher capacity link (L1) while SMCC has lower performance in the lower capacity link (L3). We also observe that SMCC presents

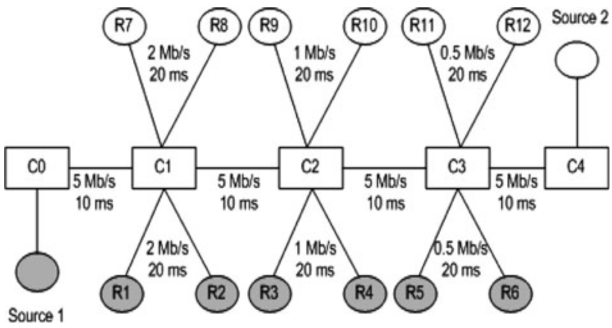

Fig. 27 Network topology for intra-fairness simulation scenario

higher bandwidth requirements than ASSP in the link connecting C0 to C1 for the transmission of the three streams. SMCC calculates the maximum transmission rate of the different multicast streams based on the transmission rate of the base layer. Under SMCC, the maximum cumulative sending rates are defined as follows:

$$B_i = 2^i \cdot B_0 \quad \text{for } i \geq 1 \tag{12}$$

where, B_0 is the transmission rate of the base layer and B_i is the transmission rate of layer i. SMCC's performance is strongly coupled with the initial setting of the base layer. To further investigate the impact of the initial rate for the base layer in SMCC we run various simulations and summarize the results in Table 4 (all values in Mb/s). We measure the bandwidth requirements for the three multicast streams in the link connecting routers C0 and C1. SMCC bandwidth consumption depends on the transmission rate of the base layer and varies from 6.49 to 7.4 Mb/s. Therefore, it is important for SMCC to choose an optimal value for the base layer during the initiation of the multicast session. This area requires a deeper investigation on how to define such algorithm.

ASSP, on the other hand, depends on the settings for the upper and lower thresholds of the simulcast streams. The upper and lower thresholds however, simply map the capacity of the bottleneck links in the path between the sender and the groups of multicast receivers. This seems to be a simple task if we consider that receivers can inform the sender about their receiving capabilities (depending on their access link for example) during the initiation of the multicast session. A low complexity algorithm such as the one presented in [41] can define the optimal number of simulcast streams that have to be transmitted in order to satisfy the majority of receivers in the multicast group. This is however left as a subject for future work. The measured Jain's fairness index for both protocols in all the bottleneck links with competing TCP flows is 0.99. Therefore, ASSP and SMCC proved to be TCP-friendly and almost equally share networks resources with the TCP flows.

7.3 Intra-Protocol Fairness

In this simulation scenario, we investigate the fairness between flows of the same protocol using the network topology in Fig. 27. In this scenario, two multicast

Table 3 Network utilization statistics

		Average throughput (Mb/s)	Link utilization (%)	TCP Average throughput (Mb/s)	BW requirements between C0 and C1 (Mb/s)
SMCC	L1	4.28	90.6	4.78	7.4
	L2	2.28	93.2	2.38	
	L3	0.84	92.5	1.01	
ASSP	L1	3.88	84.4	4.56	7.1
	L2	2.37	92.8	2.27	
	L3	0.92	97.0	1.02	

Table 4 SMCC link utilization under different transmission rates for the base layer

Base layer (Mb/s)		Average throughput (Mb/s)	Link utilization (%)	BW requirements between C0 and C1 (Mb/s)
BL = 0.5	L1	3.69	73.8	6.49
	L2	1.96	78.4	
	L3	0.84	84.0	
BL = 1.0	L1	4.28	90.6	7.4
	L2	2.28	93.2	
	L3	0.84	92.5	
BL = 2.0	L1	4.30	86.0	6.81
	L2	1.67	66.8	
	L3	0.84	84.0	

Table 5 Intra-protocol fairness simulation statistics

Link capacity	ASSP average throughput (Mb/s)		Jain's index	SMCC average throughput (Mb/s)	Jain's index
L1 (2 Mb/s)	Source 1	0.684	0.999	0.379	0.991
	Source 2	0.671		0.454	
L2 (1 Mb/s)	Source 1	0.346	0.994	0.379	1.0
	Source 2	0.403		0.379	
L3 (0.5 Mb/s)	Source 1	0.234	0.999	0.240	0.999
	Source 2	0.243		0.235	

senders transmit in opposite directions to different user groups. We set the transmission rate for the base layer of the SMCC to 250 Kb/s in order to scale better with the lower capacity link of 0.5 Mb/s. We observe that both ASSP and SMCC present high levels of intra-fairness as the available bandwidth in the bottleneck links is fairly distributed between the two multicast sources (Table 5). The Jain's fairness index in all cases is above 0.99. However, the two protocols present

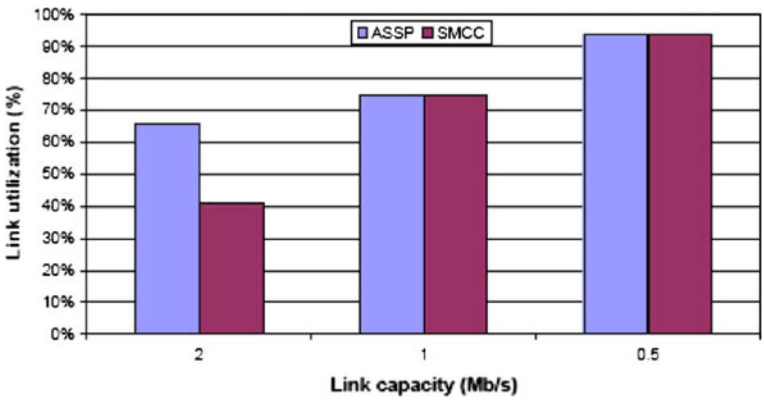

Fig. 28 Link utilization in intra-fairness scenario

different levels of link utilization. ASSP seems to show better exploitation of the available bandwidth in the bottleneck link than SMCC (Fig. 28).

7.4 Delay and Packet Loss Ratio Measurements: Minimizing Oscillations

We conduct a number of simulations with different RTTs in the network topology of Fig. 24. Both SMCC and ASSP have similar performance in terms of the average delay although ASSP outperforms SMCC in all cases (Table 6). The highest average delay is observed in SMCC, with an RRT value of 256 ms. However, this delay can be smoothed with an appropriate buffer when streaming media applications are the applications of interest. Our assessment is that both protocols present high performance with delay values that are within acceptable rates. Although multimedia applications are known as “packet loss tolerant applications”, we cannot underestimate the effects of packet losses on any application type. Therefore, packet loss ratio is an important attribute for any multimedia transmission protocol. We observe that, ASSP presents higher loss ratio than SMCC in all cases, although the values are very low for both protocols. The reason behind ASSP’s lower performance is the higher feedback intervals of the RTCP

Table 6 Simulation statistics over different RTTs

	RTT (msec)	Average end-to-end delay (msec)	Loss ratio (%)
SMCC	64	91	0.08
	128	116	0.05
	256	169	0.03
ASSP	64	86	0.2
	128	100	0.16
	256	157	0.2

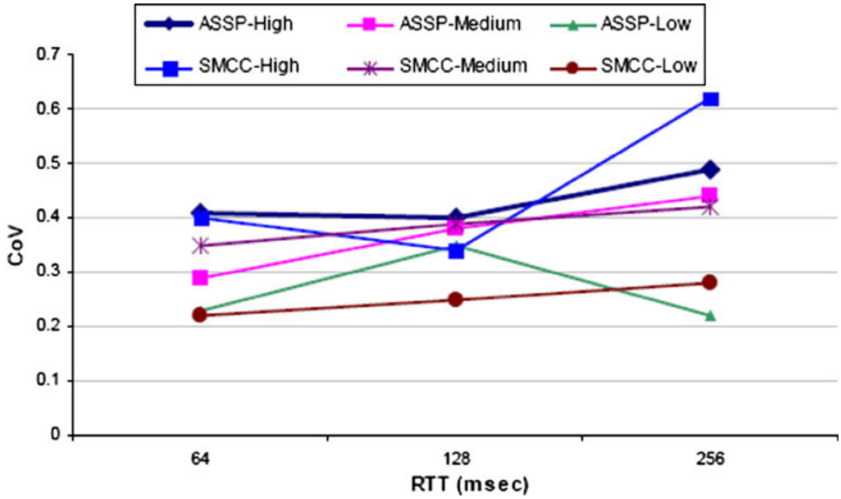

Fig. 29 CoV of average throughput over different RTTs

protocol compared to those of the TFMCC. TFMCC's sender needs to receive one feedback report from the group representative in every RTT, while ASMP's feedbacks are regulated by the RTCP protocol with higher intervals than TFMCC's.

To measure the stability of the two protocols we use the coefficients of variation (CoV) of the throughput values over the different RTTs and plot the results in Fig. 29. We observe that SMCC presents a higher level of oscillations than ASSP when the RTT values increase.

8 Conclusions: Future Work

In this paper we have presented a proposal for simulcast video transmission over heterogeneous networks. With the use of simulcast transmission, we avoided the extra overhead of more sophisticated scalable encoding.

We used an integrated simulation environment for the evaluation of our proposal. The performance evaluation metrics were a combination of "classic" network metrics with both objective and subjective video quality metrics (PSNR, MOS). This joint evaluation method provided a better understanding of the correlation and the dependencies between network metrics and the effects on the perceived video quality by the end user.

ASSP presented high stability, with minimal oscillations due to delay and bandwidth variances in the network. The internal functions of ASSP minimized join and leave requests. The low frame loss ratio in all simulation scenarios indicated that the underlying congestion control algorithm in ASSP was able to quickly detect and handle upcoming congestion caused by competing traffic.

Fairness is an important aspect of any transport protocol in that traffic should be equally shared and distributed between applications in a network. ASSP proved to be TCP-friendly, as TCP traffic enjoyed a high level of fairness when competing with ASSP traffic. This is an important attribute of ASSP as most popular user applications today are TCP-based. However, the smooth and steady behavior of ASSP results in lower bandwidth utilization when compared to TCP throughput. However, there has been always a trade-off between smooth transmission rates and high bandwidth utilization.

Uncontrolled transmission leads to network congestion and engages the retransmission of TCP packets due to packet losses. This is an undesirable situation as the retransmission of TCP packets wastes network resources. Intra-fairness of ASSP was very high, almost reaching the absolute fairness of 1.0 in accordance with the Jain's fairness index.

A direct effect of packet losses for a video application is PSNR degradation. We observed the lower PSNR values in the same network topology when ASSP competed against UDP uncontrolled traffic. The higher the packet loss ratio in a video transmission was, the lower the obtained PSNR values were and hence the lower the level of satisfaction at the end user was.

We compared our solution against a well-defined congestion control scheme for cumulative layer multicast. SMCC performance was highly coupled to the initial settings of the base layer, while ASSP needed to know in advance the available bandwidth of user groups behind the same bottleneck link. Both protocols proved to be TCP-friendly as equally shared network resources with TCP flows. The Jain's fairness index was 0.99 in all simulation scenarios. ASSP outperformed SMCC in terms of link utilization and scaled better in the high capacity links in almost all of the simulation sets. Delay measurements showed that both protocols had a good performance while ASSP was more efficient than SMCC. SMCC responded better to network congestion than ASSP whose feedback functions depended on the high intervals of the RTCP protocol. As a result, ASSP was "slower" than SMCC in responding to network congestion and presented higher loss ratio. Simulations with different RTTs showed that ASSP seemed to be more stable than SMCC.

A deeper investigation is needed into how to define the algorithms for optimal performance for both protocols. A possible solution for SMCC would be on-the-fly selection of the transmission rate for the base layer to deal with unpredictable and dynamic changes to network conditions. On the other hand, ASSP should also be able to optimize the number of the available streams based on network conditions.

A more complete performance evaluation, based not only on network-centric metrics but also on specific multimedia-based quality metrics would provide a better insight on the advantages and drawbacks of the two solutions tested here. It would be also interesting to evaluate the performance of ASSP in wireless scenarios in which the cause of packet losses is not always a congested link. Furthermore, our future work includes the implementation of automatic tuning of ASSP parameters in order to optimize protocol's performance under various network conditions. Finally we plan to implement a prototype of ASSP using Java and JMF (Java Media Framework). All the above areas are left for future work.

References

1. RFC 4654: On TCP-friendly multicast congestion control (TFMCC). In: Widmer, J., Handley, M. (eds.), August (2006)
2. Yousefi'zadeh, H., Jafarkhani, H.: Achieving inter-receiver fairness utilizing layered media multicast control (LMMC). In: Proceedings of the IEEE International Conference on Communications, ICC 2004, vol. 2, pp. 1151–1155, Paris, France, June (2004)
3. RFC 3550: RTP: a transport protocol for real-time applications. In: Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V. (eds.) July (2003)
4. Bouras, C., Gkamas, A., Kioumourtzis, G.: Adaptive smooth simulcast protocol for multimedia transmission. In: The Fourteenth IEEE Symposium on Computers and Communications (ISCC09), Sousse, Tunisia, 5–8 July (2009)
5. Bouras, C., Gkamas, A., Kioumourtzis, G.: Adaptive smooth multicast protocol for multimedia data transmission. In: 2008 International Symposium on Performance Evaluation of Computer and Telecommunication Systems—SPECTS 2008, Edinburgh, UK, 16–18 June (2008)
6. Bouras, C., Gkamas, A., Kioumourtzis, G.: Comparison of single-rate multicast congestion control protocols vs. ASMP. In: 16th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)—MASCOTS 2008, Baltimore, MD, USA, 8–10 Sept (2008)
7. Rizzo, L.: pgmcc: A TCP-friendly single-rate multicast congestion control scheme. In: Proceedings of ACM SIGCOMM '00, Stockholm, Sweden, 28 Aug–1 Sept (2000)
8. RFC 4605: Internet group management protocol (IGMP)/Multicast. In: Fenner, B. et al. (ed.) August (2006)
9. Kwon, G.-I., Byers, J.: Smooth multirate multicast congestion control. In: Proceedings of IEEE INFOCOM, San Francisco USA, 30 March–1 April (2003)
10. McCanne, S., Jacobson, V., Vetterli, M.: Receiver-driven layered multicast. In: Proceedings of ACM SIGCOMM, Stanford University, California, USA, 26–27 August (1996)
11. Legout, A., Biersack, E.: PLM: fast convergence for cumulative layered multicast transmission schemes. In: Proceedings of ACM SIGMETRICS, Santa Clara, California, USA, 17–21 June (2000)
12. Byers, J.W. et al.: FLID-DL congestion control for layered multicast. In: Proceedings of NGC, Stanford University, Palo Alto, California, USA, 8–10 November (2000)
13. Puangpronpitag, S., Boyle, R., Djemame, K.: Performance evaluation of layered multicast congestion control protocols: FLIDDL vs. PLM. In: Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 03), Montreal, Canada, July (2003)
14. The UCB/LBNL network simulator, software online: http://www.isi.edu/nsnam/ns/
15. RFC 5166, Floyd, S.: Metrics for the evaluation of congestion control mechanisms. Network Working Group, March (2008)
16. Shi, S., Waldvogel, M.: A rate-based end-to-end multicast congestion control protocol. In: Proceedings of Fifth EEE Symposium on Computers and Communications (ISCC 2000), Antibes-Juan les Pins, France, 3–6 July (2000)
17. Byers, J., Luby, M., Mitzenmacher, M.: Fine-grained layered multicast. Proc. INFOCOM 2001 **2**, 1143–1151 (2001)
18. Byers, J., Kwon, G.: STAIR: practical aimed multirate multicast congestion control. In: Proceedings of NGC, London UK, 7–9 November (2001)
19. Li, X., Ammar, M., Paul, S.: Video multicast over the internet. IEEE Netw Mag **13**(2), 46–60 (1999)
20. Jiang, T., Zegura, E.W., Ammar, M.H.: Inter-receiver fair multicast communication over the Internet. In: Proceedings of NOSSDAV'99, AT&T Learning Center, Basking Ridge NJ, USA, 23–25 June (1999)
21. Bouras, C., Gkamas, A.: SRAMT-S: A hybrid sender and receiver-based adaptation scheme for TCP friendly multicast transmission using simulcast approach. In: 1st IFIP Workshop on Internet Technologies, Applications and Social Impact (WITASI-02), Wroclaw, Poland, pp. 105–122, 10–11 October (2002)
22. Pandhye, J., Kurose, J., Towsley, D., Koodli, R.: A model based TCP-friendly rate control protocol. In: Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Basking Ridge, NJ, USA, 23–25 June (1999)

23. ITU-T Recommendation G.100/P.10 Amendment 1: New appendix. I—Definition of quality of experience (QoE). Jan (2007)
24. ITU-T Recommendation P.910: Subjective video quality assessment methods for multimedia applications. April (2008)
25. Rohaly, M. et.al.: Video quality experts group: current results and future directions. In: SPIE Visual Communications and Image Processing, Perth, Australia, vol. 4067, pp. 742–753, 21–23 June (2000)
26. ITU-T Recommendation G.1080: Quality of experience requirements for IPTV services. Dec (2008)
27. Riley, M.J., Richardson, I.E.G.: Digital Video Communications. Artech House, 685 Canton Street, Norwood (1997)
28. ITU-R Recommendations BT.500-11. Methodology for the subjective assessment of the quality of television pictures (2002)
29. Klaue, J., Rathke, B., Wolisz, A.: EvalVid—A framework for video transmission and quality evaluation. In: Proceedings of the 13th International Conference on Modeling, Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois (2003)
30. Lie, A., Klaue, J.: Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR Video. ACM/Springer Multimedia Systems Journal autumn (2007)
31. FFmpeg audio/video conversion tool, http://sourceforge.net/projects/ffmpeg/ (information available online—visited 08/2009)
32. Tionardi, L., Hartanto, F.: The use of cumulative inter-frame for adapting video transmission rate. In: Proceedings of TENCON, Bangalore, India, 15–17 October (2003)
33. Jain, R., Chiu, D., Hawe, W.: A quantitative measure of fairness and discrimination for resource allocation in shared systems. DEC Research Report TR-301, Tech. Rep., Sept (1984)
34. ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6: Joint Draft 11: Scalable Video Coding. Joint Video Team (JVT) of ISO/IEC MPEG&ITU-T VCEG, Document JVT-X201, Geneva, Switzerland, July (2007)
35. Kim, T. Ammar, M. H.: A comparison of layering and stream replication video multicast schemes. In Proceedings of NOSSDAV'01, Port Jefferson, New York USA, 25–26 June (2001)
36. de Cuetos, P., Saparilla, D., Ross, K.W.: Adaptive streaming of stored video in a TCP-friendly context: multiple versions or multiple layers. In: Proceedings of Packet Video Workshop, Apr (2001)
37. Avramova, Z. et al.: Comparison of simulcast and scalable video coding in terms of the required capacity in an IPTV network. In: Proceedings of PV 2007, Lausanne, Switzerland, November (2007)
38. Smooth Multirate Multicast Congestion Control, (information available online—visited 08/2009), on line: http://cs-people.bu.edu/guin/smcc.html
39. Research Unit 6//Research Academic Computer Technology Institute activities on multimedia transmission (information available online—visited 08/2009), on line: http://ru6.cti.gr/ru6/ns_rtp_home.php
40. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP reno performance: a simple model and its empirical validation. IEEE/ACM Trans Netw **8**(2), 133–145 (2000)
41. Liu, J., Li, B.: Optimal stream replication for video simulcasting. In: 10th IEEE International Conference on Network Protocols, Paris, France, 12–15 November (2002)

Author Biographies

Christos Bouras obtained his Diploma and PhD from the Computer Science and Engineering Department of Patras University (Greece). He is currently a Professor in the above department. Also he is a scientific advisor of Research Unit 6 in Research Academic Computer Technology Institute (CTI), Patras, Greece. His research interests include Analysis of Performance of Networking and Computer Systems, Computer Networks and Protocols, Telematics and New Services, QoS and Pricing for Networks and Services, e-learning, Networked Virtual Environments and WWW Issues. He has extended professional experience in Design and Analysis of Networks, Protocols, Telematics and New Services. He has published more than 350 papers in various well-known refereed conferences and journals. He is a co-author of 8 books in Greek. He has been a PC member and referee in various international journals and conferences. He has participated in R&D projects such as RACE, ESPRIT, TELEMATICS, EDUCATIONAL MULTIMEDIA, ISPO, EMPLOYMENT, ADAPT, STRIDE, EUROFORM, IST, GROWTH and others. Also he is member of experts in the Greek Research and Technology Network (GRNET), Advisory Committee Member to the World Wide Web Consortium (W3C), IEEE-CS

Technical Committee on Learning Technologies, IEEE ComSoc Radio Communications Committee, IASTED Technical Committee on Education W 6.4 Internet Applications Engineering of IFIP, ACM, IEEE, EDEN, AACE, New York Academy of Sciences and Technical Chamber of Greece.

Apostolos Gkamas obtained his Diploma, Master Degree and PhD from the Computer Engineering and Informatics Department of Patras University (Greece). He is currently Scientific cooperator at the Research Unit 6 of the Research Academic Computer Technology Institute, Patras, Greece, visiting lecturer in Hellenic Open University in the Informatics Curriculum and Visiting Assistant Professor in the Informatics Department of University of Ioannina. His research interests include Computer Networks, Telematics, Multimedia transmission and Cross Layer Design. More particular he is engaged in transmission of multimedia data over networks and multicast congestion control. He has published more than 60 papers in international Journals and well-known refereed conferences. He is also co-author of three books (one with subject Multimedia and Computer Networks one with subject Special Network Issues and one with subject IPv6). He has participated in various R&D project (in both EU and national) such as IST, FP6, FP7, Intereg eLearning, PENED, EPEAEK, Information Society.

Georgios Kioumourtzis is a PhD candidate in the Computer Engineering and Informatics Department of Patras University (Greece). He is currently a Lieutenant Colonel in the Greek Army. He received his BS from the Hellenic Military Academy and graduated in 1996 from the School of Telecommunications for Signal Officers. In 2005, he received the Master of Science in System's Engineering and the Master of Science in Computer Science from the Naval Postgraduate School (NPS) California, USA. His thesis work was related to Mobile Ad-Hoc Wireless Networks (MANETs). His current research interests include Multimedia transmission over heterogeneous networks, transmission protocols, cross-layer optimization in wireless networks and IEEE 802.11x technology.