INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2012; **22**: 349–372
Published online 7 February 2012 in Wiley Online Library (wileyonlinelibrary.com) DOI: 10.1002/nem.810

TFMCC versus ASMP: lessons learned from performance evaluation

Georgios Kioumourtzis,[1] Christos Bouras[2,*†] and Apostolos Gkamas[2]

[1]*Department of Computer Engineering and Informatics, University of Patras, 26500 Rion, Patras, Greece*
[2]*Research Academic Computer Technology Institute, University of Patras, 26500 Rion Patras, Greece*

SUMMARY

In this article we present a simulation-based comparison of one of the best-known multicast congestion control schemes—TCP-friendly Multicast Congestion Control (TFMCC)—against our proposed Adaptive Smooth Multicast Protocol (ASMP). ASMP consists of a single-rate multicast congestion control mechanism which takes advantage of the RTCP Sender (SR) and Receiver Reports (RR) in order to adjust the sender's transmission rate in respect of the network conditions. The innovation in ASMP lays in the 'smooth' transmission rate, which is TCP-friendly and prevents oscillations. We use an integrated simulation environment named Multi-Evalvid-RA for the evaluation of the two congestion control schemes. Multi-Evalvid-RA provides all the necessary tools to perform simulation studies and assess video quality by using both network-centric metrics along with video quality measurements. Performance evaluation results show that ASMP is a very efficient solution for rate-adaptive multimedia applications and a serious competitor to well-known TFMCC. Copyright © 2012 John Wiley & Sons, Ltd.

Received 23 November 2010; Revised 15 November 2011; Accepted 15 November 2011

KEY WORDS: Multimedia transmission, TCP Friendly, Media Friendly, Congestion Control, Multicast

1. INTRODUCTION

Multicast transmission is a preferable solution of group communication applications such as multimedia applications, information dissemination services, software upgrade services, etc. However, design complexity is very high when trying to accommodate a transport protocol for multimedia data transmission in the multicast domain. The main reason is that multimedia applications pose their own constraints and quality of service (QoS) requirements, as a direct result of their nature. These applications are different from Transmission Control Protocol (TCP)-based applications and characterized mainly by the following three properties:

- demand for high data transmission rates (bandwidth-consuming applications);
- sensitivity to packet delays (latency and jitter); and
- tolerance to packet losses (packet-loss tolerant applications).

Congestion control and TCP-friendliness[1] pose additional design requirements as shark tooth-like transmission rates may be too difficult to be followed by audio–video (AV) encoders and decoders. TCP congestion control produces high fluctuations in the transmission rate which are not suitable for the current AV codecs, which expect predictive and stable bandwidth allocation. In general, TCP is

*Correspondence to: Christos Bouras, Research Academic Computer Technology Institute and University of Patras, N. Kazantzaki St, University of Patras, 26500 Rion, Greece.
†E-mail: bouras@cti.gr

[1]*We define as TCP-friendly flow one that consumes no more bandwidth than a TCP connection, which is traversing the same path with that flow.*

acceptable for multimedia transmission if the average transmission rate is twice as much as the average bit rate (i.e. bandwidth requirements) of the bit stream [1]. Multimedia transmission can cope with transient fluctuations of the transmission rate with the use of buffers at the clients. However, a long initial data pre-fetch in a buffer before the player starts playing the stream is not easily accepted by all end users. Moreover, in real-time video applications and conversational media large pre-fetch buffers are not acceptable. For multimedia applications smooth and steady transmission rates with low delay are more important attributes than guaranteed and on-order delivery of data packets.

Therefore, our motivation is to design a multicast congestion control scheme for multimedia transmission (we call this control scheme Adaptive Smooth Multicast Protocol (ASMP)) that is TCP-friendly and at the same time meets the QoS requirements posed by multimedia applications. Our main concept is to exploit the functionality of an existing and widely used protocol in order to obtain the necessary feedback reports and calculate the network related metrics for an adaptive and TCP-friendly behavior. We choose Real-time Transport Protocol (RTP) [2] and its associate Real-time Transport Control Protocol (RTCP), which is the *de facto* standard for multimedia transmission in networks today. RTCP employs feedback suppression algorithms that increase scalability. It is worth mentioning that ASMP does not require any additional support from the routers or the underlying IP-multicast protocols. This property allows easy deployment over unmanaged networks like the Internet.

We focus in this article on a detailed evaluation, which is based on both network-centric along with video-quality metrics. This 'joint' performance evaluation process provides the evaluation platform to better understand the benefits and limitations of our proposal against the well-known TCP-friendly Multicast Congestion Control (TFMCC) [3]. A short initial presentation with simpler simulation scenarios and performance evaluation results is presented in [4]. The performance evaluation results show that ASMP is a very efficient solution for rate-adaptive multimedia applications and a serious competitor to TFMCC.

The rest of this paper is organized as follows. Section 2 discusses related work. We provide an overview of the two congestion control schemes in Section 3. In Section 4 we discuss the latest video quality assessment methods. Section 5 presents the simulation environment, which combines ns-2 [5] with an integrated tool-set for performance evaluation of video transmission. In Section 6 we present the performance evaluation of ASMP against TFMCC and the related results. We conclude our paper in Section 7. Finally, we discuss future work in Section 8.

2. RELATED WORK

Research work in the area of video multicast transmission can be classified into two main categories in respect of the number of transmitted layers that are involved during transmission:

- *Single-layer design*: in this category the sender transmits a single layer and the transmission rate is defined by the receiver with the lowest bandwidth capacity.
- *Multi-layer design*: under this approach, video is transmitted by a number of different layers (streams) and each individual receiver joins the multicast stream(s) that is closer to its bandwidth capabilities.

In this paper we focus on the field of single-rate design. The discussion of the limitations of single-rate multicast protocols versus multi-rate protocols is beyond the scope of this paper.

Up to now there have been promising approaches in the single-rate area. TFMCC [3,6] extends the basic mechanisms of TFRC [7] to support single-stream multicast congestion control. The most important attribute of TFMCC is the suppression of feedback receiver reports. TFMCC uses the receiver with the lowest receiving capacity acting as the representative of the multicast group. The sender adjusts the transmission rate based on feedback reports from the group representative. PGMCC [8] is a window-based TCP scheme that is based on positive ACKs between the sender and the group representative (*acker*). Only the *acker* is tasked to send positive ACKs to the sender, mimicking the 'classic' TCP receiver. All other receivers in the multicast session send NACKs whenever they discover packet losses. TBRCA [9] targets at maximizing the overall amount of multimedia data to the whole set of receivers and at the same time serves receivers with low bandwidth connections. With the use of a bandwidth rate

DOI: 10.1002/nem

control algorithm, it dynamically controls the output rate of the video coder. LDA + [10] is an additive increase and multiplicative decrease (AIMD) algorithm in which the addition and reduction values are dynamically determined based on current network conditions. To do so, LDA + uses feedback from the receivers based on the RTP protocol. LDA + employs the TCP analytical model in order to estimate a TCP-friendly bandwidth share in the event of packet losses. LDA + does not implement any additional feedback suppression mechanism except for the one that is provided by the RTCP protocol. MDP-CC [11] uses also representatives for the adaptation of the transmission rate by the sender. MDP-CC maintains a pool of representative candidates for the representative selection. ERMCC [12] implements a congestion control scheme that is based on a new metric named TRAC (Throughput Rate At Congestion). ERMCC can be implemented only by the sender and the receivers of the multicast group without any network support. The sender dynamically selects one of the slowest receivers as Congestion Representative (CR), and only considers its feedback reports for the adaptation of the transmission rate. The feedback suppression philosophy is similar to that of TFMCC, although it seems to offer higher scalability.

TFMCC has been compared against other possible solutions [13–16] 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 were purely based on ‘classic’ network metrics (e.g. throughput, packet loss ratio, delay jitter). However, network metrics cannot characterize the quality of the resulting video transmission and may lead to debatable results because the perceived video quality at the end user is not measured. It is also very difficult to transform or correlate network metrics into quality of experience (QoE) [17] metrics of a video transmission. To the best of our knowledge we have not seen a similar evaluation of TFMCC written by other authors based on video-centric metrics.

The above limitations, as part of the simulation environment, undermine the performance evaluation process in which quality measurements for multimedia data transmission (e.g. peak signal-to-noise ratio (PSNR), mean opinion score (MOS)) are missing. Different video 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. In addition, small variation of network metrics (e.g. packet loss ratio) may have an important effect on video quality metrics (e.g. PSNR). Therefore, it is important to study the performance of any proposed solution by using real video files and associate the simulation results with video QoE metrics.

To overcome the above limitations the designers of Evalvid [18] provided the framework and tools to perform several studies of video transmission over a real or simulated network. Under this new framework the performance evaluation of the video transmission can include QoE metrics that characterize the perceived video quality by the end user. The early version of Evalvid was restricted to non-adaptive video transmission in which the video file was transmitted by a constant bit rate (CBR) video source. A later work named Evalvid-RA [19] extended the Evalvid by adding rate adaptive video transmission functionality. However, Evalvid-RA was also restricted to unicast transmission and therefore simulations and performance evaluation studies with multicast protocols were excluded.

For the purpose of this work we extended the Evalvid-RA to Multi-Evalvid-RA in order to be able to support multicast transmission of multimedia data. Multi-Evalvid-RA integrates the extensions made to ns-2 in our previous work [20] and provides a basic API for the implementation of any rate-adaptive scheme within or on top of RTP/RTCP protocols. By doing so, any congestion control mechanism can be integrated into Multi-Evalvid-RA by making use of either the feedback mechanism of RTP/RTCP or any other preferable solution.

3. MULTICAST CONGESTION CONTROL SCHEMES

In this section we briefly discuss the functionality of TFMCC and ASMP.

3.1. TCP-friendly Multicast Congestion Control

TFMCC is a single-rate multicast congestion control scheme that extends TFRC from the unicast to multicast domain. The design goals of TFMCC are to provide a multicast congestion control that is

DOI: 10.1002/nem

TCP-friendly, highly responsive to network changes and also suitable for multimedia data transmission. One important attribute of TFMCC is the feedback suppression algorithm in which only the receiver with the lowest receiving capacity, which is termed the current-limiting receiver (CLR), sends frequent feedback reports. The sender adjusts its transmission rate based on CRL feedback reports. The rest of the receivers in the multicast group send their feedback at longer time intervals in order to prevent feedback implosion phenomena at the sender side. TFMCC's TCP-friendly bandwidth share is measured by receivers with the use of the following TCP equation:

$$r_{\mathrm{tcp}}^{i} = \frac{P}{t_{\mathrm{RTT}}\left(\sqrt{\frac{2l}{3}} + \left(12\sqrt{\frac{3l}{8}}\right)l(1+32l^2)\right)} \tag{1}$$

where, r_{tcp}^{i} is the receiver's i estimation (in bytes/s), P is the packet size in bytes, l is the packet loss ratio and t_{RTT} is the round trip time (RTT) of the link between the sender and the receiver. The long-term TCP-friendliness is defined in TFMCC's specification to be 'no more than twice the sending rate of a TCP flow which is traversing the same link as the TFMCC flow'. For the time being, TFMCC is under the review of the Internet community as an experimental RFC [3].

3.2. *Adaptive Smooth Multicast Protocol*

ASMP consists of a single-rate multicast congestion control scheme, which takes advantage of the RTCP Sender (SR) and Receiver Reports (RR). The receiver emulates the behavior of a TCP agent and, as such, when packet losses occur it estimates a TCP-friendly bandwidth share r_{tcp}^{i} in every RTCP report interval, with the use of the following analytical model [21]:

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

where, r_{tcp}^{i} is the receiver's i estimation (in bytes/s), P is packet size in bytes, l is the packet loss rate, t_{out} is the TCP retransmission timeout, t_{RTT} is the RTT 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 times 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_{tcp}^{i} must not be increased more than one P/RTT.

The innovation in ASMP is that the calculation of the sender's transmission rate is performed in such a way that oscillations are reduced in a smooth fashion. We combine this attribute with the long-term TCP-friendliness, meaning that the multimedia stream consumes no more bandwidth than a TCP connection, which is traversing the same path as the multimedia stream. Moreover, with the use of RTCP feedback reports ASMP provides better scalability, as the amount of feedback reports is controlled by the RTCP protocol and they cannot exceed a specified threshold, expressed as a percentage of the total available bandwidth [2]. Without disseminating any additional feedback reports (ACKs or NACKs) other than those of RTCP sender and receiver reports, ASMP increases bandwidth utilization for user data. Therefore, we define the following function that provides smoother transmission rate estimations:

$$r_{\mathrm{tcp}}^{i} \leftarrow r_{\mathrm{tcp}}^{\mathrm{inst}}\cdot\gamma + (1-\gamma)\cdot r_{\mathrm{tcp}}^{i} \tag{3}$$

where $r_{\mathrm{tcp}}^{\mathrm{inst}}$ is the latest estimation of the transmission rate measured by the receiver and γ is the *smoothness factor*, with values between 0 and 1:

$$0 \leq \gamma \leq 1 \tag{4}$$

The value of γ is not static but depends on the level of congestion in the network. Unfortunately, packet loss or RTT on their own cannot provide a clear picture of the network congestion level, and when packet losses occur the network is already congested. Thus to avoid packet losses we need to implement an

DOI: 10.1002/nem

early-warning congestion algorithm, in which—in the ideal case—the receiver can detect upcoming network congestion prior to any packet losses. To do so, we define *congestion indicators* (CI), which provide the warnings of the upcoming congestion. An effective way to assess the level of congestion is to compare the *long running average* of cumulative jitter [22] against a *short running average*. When the short running average is bigger than the long running average the congestion level is increasing, which means that we meet the state CONGESTED. Conversely, when the short running average is smaller than the long running average we asses that the network is UNLOADED. Therefore, the *smoothness factor* regulates the behavior of the congestion control mechanism, by making it less or more aggressive, in respect of the level of congestion in the bottleneck link. In this way, ASMP succeeds in having a congestion control that is 'smooth', without suffering from high oscillations, and at the same time well responsive to network changes. We accept that the network can be on one of the following states:

- *State CONGESTED*: in this state cumulative jitter delay has increasing values over time as jitter delay is also increasing over time. Therefore, we need to adjust the sender's transmission rate in such way as to prevent congestion in the bottleneck link that will lead to packet losses. In order to do so we should give a low value to γ, so that we can regulate the increase in the transmission rate and make it less than one packet per RTT. Otherwise, if the sender continues to increase the transmission rate at a constant rate by one P/RTT it will soon cause congestion and packet losses.
- *State UNLOADED*: in this state cumulative jitter has an almost constant value. Parameter γ should have values close to one to allow a constant increase of the transmission rate that is close to one P/RTT.

For our simulations that are presented later in this paper we define the following values of parameter γ based on simulation results for the two different network states:

$$\begin{array}{l} \text{UNLOADED} \rightarrow \gamma = 0.9 \\ \text{CONGESTED} \rightarrow \gamma = 0.2 \end{array} \tag{5}$$

ASMP is developed on top of the RTP/RTCP protocol. This approach has several advantages because the complexity is moved up to the application layer, leaving the operating system and network elements untouched as well. With the use of RTCP feedback reports we provide better scalability as the number of feedback reports are controlled by the RTCP protocol and they cannot exceed a specified threshold, expressed as a percentage of the total available bandwidth [2]. Without disseminating any additional feedback reports (ACKs or NACKs) than those of RTCP sender and receiver reports, ASMP increases bandwidth utilization for user data.

A high-level overview of the functionality of ASMP is presented below:

- The receiver measures the loss event ratio based on RTP packet sequence numbers.
- The sender measures the RTT between itself and the receiver based on receiver RTCP feedback reports, and transmits the measured RTT to this receiver with the use of the extension mechanisms of the RTP/RTCP application (APP)-specific part.
- The receiver measures a TCP-friendly bandwidth share with the use of the TCP analytical model.
- The receiver calculates a new smooth transmission rate based on the measured value of the '*smoothness factor*'.
- The receiver sends the calculated smooth rate to the sender by making use of the extension mechanisms of the RTP/RTCP application (APP)-specific part.
- The sender adjusts its transmission rate based on RTCP feedback receiver reports.

More details on ASMP can be found in Bouras *et al*. [23].

4. VIDEO QUALITY ASSESSMENT METHODS

In this paragraph we discuss video quality assessment methods and metrics that are used for the evaluation of ASMP. What we are interested in is to measure the quality of the perceived video file by the end user. This

DOI: 10.1002/nem

is termed as quality of experience (QoE) and is defined by the ITU-T [17]. The interesting and difficult part in this process is how to choose the methods and the metrics for assessing 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 [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, reduced and 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. Although MOS is an effective way to measure the QoE of any multimedia service for a user, is considered 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) by directly comparing the video file sent by the sender with the same file at the end user on a frame-by-frame basis. Equation (6) gives the definition of PSNR between the luminance component Y of source image S and the destination image d [18]:

$$\mathrm{PSNR}(n)_{\mathrm{dB}} = 20\log_{10}\left(\frac{V_{\mathrm{peak}}}{\sqrt{\frac{1}{N_{\mathrm{col}}N_{\mathrm{row}}}\sum_{i=0}^{N_{\mathrm{col}}}\sum_{j=0}^{N_{\mathrm{row}}}\left[Y_S(n,i,j)-Y_D(n,i,j)\right]^2}}\right) \tag{6}$$

where $V_{\mathrm{peak}} = 2^k - 1$ and k is the number of bits per pixel (luminance component).

The PSNR values of all individual video frames are then averaged to produce the mean PSNR of the complete video sequence. This is then mapped to the corresponding MOS value (Table 1). We need to point out, however, that PSNR-to-MOS mapping provides only an estimation of the perceived video quality by the end user. Other objective video quality metrics such as the psycho-visual metrics, which are based on models of the human visual system (HVS), have been proven more accurate than PSNR [27]. However, psycho-visual metrics present very high complexity compared with PSNR.

In our work, as we compare the performance of two multicast control schemes, we are mainly interested in the relative QoE and, as such, absolute metrics are not of great interest. In this case PSNR can be used as the objective quality metric to assess the performance of the two tested congestion control schemes.

We also need to point out that in general there is no a single method accepted to assess the quality of a video transmission system. Although we have discussed some aspects of the problem, an in-depth study would be beyond the scope of this paper.

Table 1. ITU-R quality and impaired scale, and PSNR to MOS mapping [18]

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

DOI: 10.1002/nem

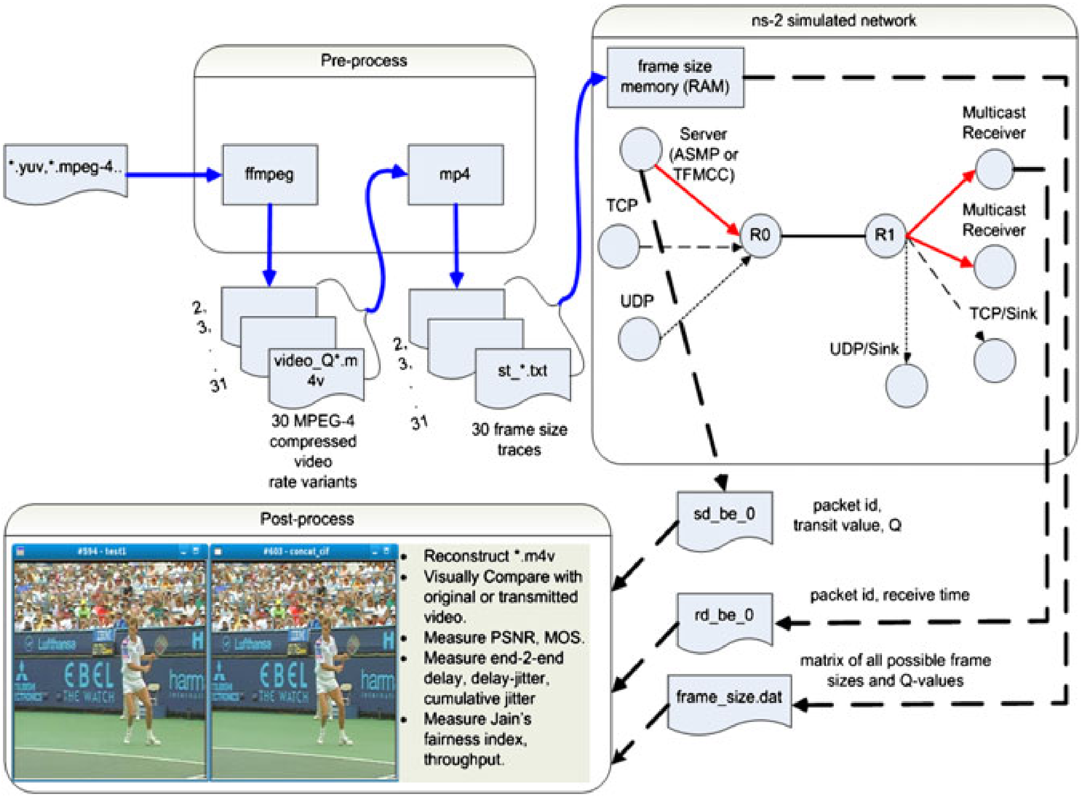

Figure 1. Overview of the Multi-Evalvid-RA: pre-processing, ns-2 simulation environment, post-processing

5. SIMULATION ENVIRONMENT

The simulation environment consists of three parts and is depicted in Figure 1. During the pre-processing a raw video file, which is usually stored in YUV format, is encoded with the desired video encoder[2] 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* [28] free video encoder for the creation of video clips. An MPEG-4 [29] encoder such us *ffmpeg* generates three types of frames. Frames are arranged in group of pictures (GOP). A GOP consists of exactly one intra-frame (I-frame), some related predictive frames (P-frames) and optionally some bidirectional frames (B-frames) between them (these I- and P-frames). I-frames have the lowest compression and contain information from encoding a still image. P-frames are encoded from the previous I- or P-frames. B-frames are encoded bidirectionally from the preceding and following I- and P-frames. B-frames are encoded with the highest compression and require the lowest transmission rates.

For our simulations, we use YUV video sequences that are publicly available [30]. We combine YUV test sequences to create video files with the desired duration and encode them into MPEG-4 video format. All the encoded MPEG-4 video clips have a temporal resolution of 25 frames per second and GOP pattern IBPBPBPBPBPBP, with a size of 12 frames. The frame size of all clips is 352 $\times$ 288 pixels, which is known as the common intermediate format (CIF). Table 2 presents an approximation between the video encoding rates per quantizer scale of the used MPEG-4 video encoder. The encoded video is then traced to produce 30 frame size trace files. We use the mp4.exe program of Evalvid to create the trace files. At the end of the pre-processing phase we have 30 m4v files with their associated frame size files. The frame size files are stored in RAM to avoid accessing external files during simulation time, which would increase the processing time.

---

[2]*Currently H.263 and MPEG-4 are supported by Evalvid.*

DOI: 10.1002/nem

Table 2. Encoding rates of video sequence

Q scale[a]	Encoding rate (kb/s)	Q scale	Encoding rate (kb/s)	Q scale	Encoding rate (kb/s)
2	1642	12	268	22	142
3	1070	13	247	23	135
4	818	14	229	24	129
5	649	15	213	25	124
6	543	16	199	26	120
7	464	17	186	27	155
8	408	18	175	28	111
9	361	19	166	29	116
10	324	20	156	30	104
11	293	21	149	31	100

[a]Q scale, quantizer scale.

The ns-2 creates the simulated network. The video file is transmitted from the server to a group of multicast receivers. During the simulation time, we store the traces of both the server and the receivers to enable the calculation of network and video performance metrics (PSNR, MOS) and reconstruction of the received video file.

The third part of the simulation environment consists of reconstruction of the transmitted video and measurement of the performance evaluation metrics. The following metrics are stored and calculated:

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

The Evaluation Trace Rate Adaptive (et_ra) program of Evalvid-RA reconstructs the received video at the end user. This program was modified to include the aforementioned delay measurements. Usage of the above tools is described in the Multi-Evalvid-RA package [31].

6. PERFORMANCE EVALUATION

The performance evaluation of transport protocols is a difficult and complex task. A first important issue is related to the different objectives that these protocols may have. A transport protocol may be optimized to maximize per-flow throughput and another optimized for low delay to satisfy the application's specific

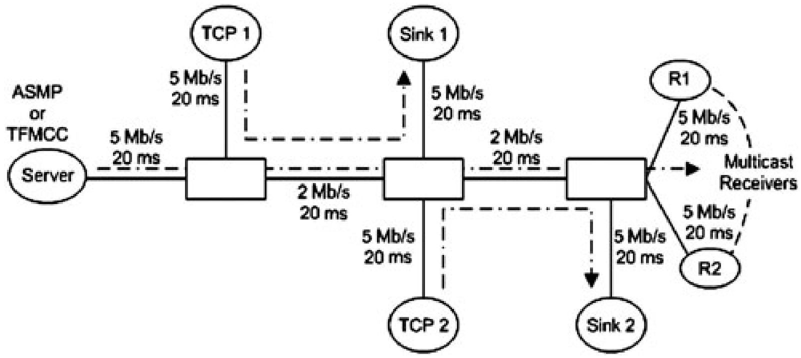

Figure 2. Parking lot bottleneck scenario

Int. J. Network Mgmt 2012; **22**: 349–372
DOI: 10.1002/nem

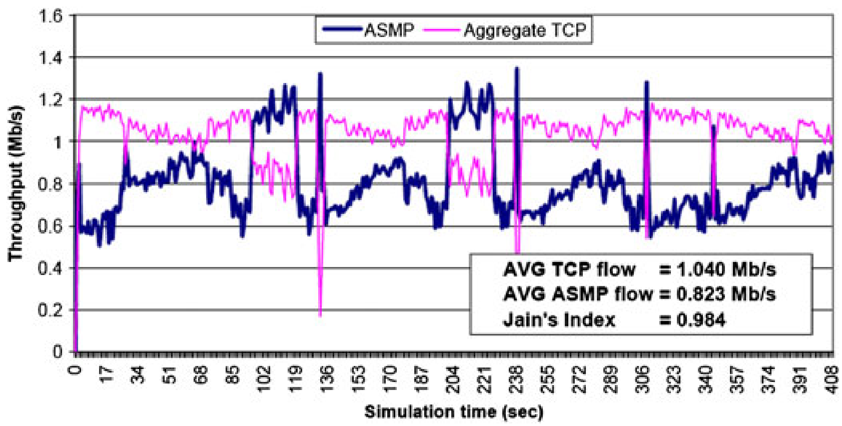

Figure 3. ASMP vs. TCP traffic

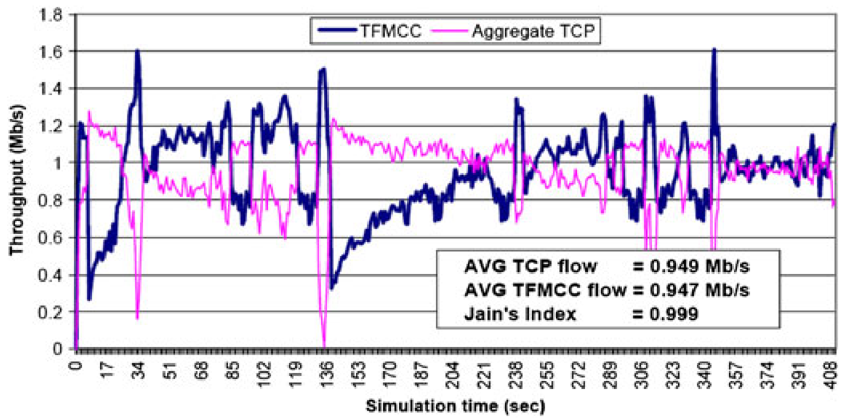

Figure 4. TFMCC vs. TCP traffic

requirements. A second important issue is related to the evaluation methods and metrics under which we investigate their performance.

The recent RFC 5166 [32] provides valuable guidance on the various aspects related to the performance evaluation of transport protocols and in particular to the congestion control mechanisms that are implemented by them. Therefore, by taking into account the RFC 5166 and also video quality indicators (PSNR and MOS), we conduct a number of experiments to investigate, among other aspects, the following:

- TCP-friendly behavior, when multicast receivers share the same bottleneck link with TCP flows;
- perceived video quality at the end user;
- effects of packet losses on video quality;
- effects on the video quality of delay constraints;
- performance of the tested protocols in scenarios with varying RTT values;
- fairness among flows of the same protocol; and finally
- responsiveness to dynamics of competing UDP flow.

6.1. TCP fairness and Jain's fairness index measurements

As in the real world there are multiple bottleneck links on a path between sender and receiver(s) we use the simulation scenario of Figure 2. In this scenario, there is a long multicast video flow with duration of 408 seconds (6.8 minutes), which represents a typical video size in social networking applications. This video sequence passes through two bottleneck links. Two additional short TCP flows are passing

DOI: 10.1002/nem

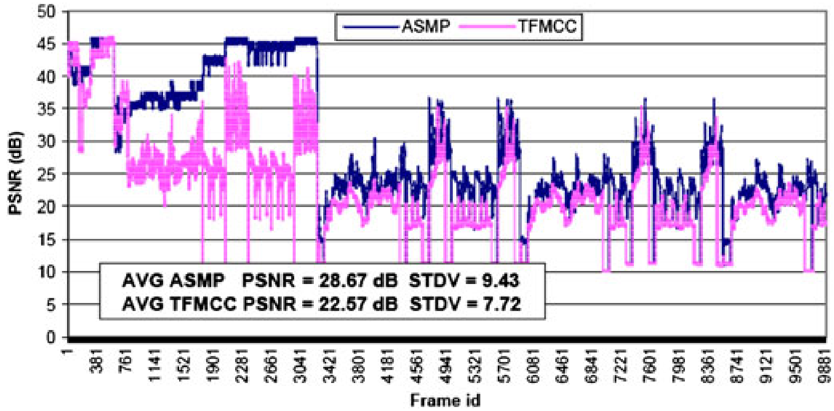

Figure 5. Resulting PSNR values (frame by frame) of TFMCC and ASMP

through only one bottleneck link. R1 to R2 stand for the video receivers. We use Drop Tail queue in the routers with a buffer size of 50 packets, which is a default value in ns-2, and set the access link capacity of all agents to 5 Mb/s with an access delay of 20 ms. Therefore, the total one-way delay on the path from the video source to video receivers is 80 ms, giving an RTT of 160 ms. The maximum transmission unit (MTU) is set to 1018 bytes for both ASMP and TFMCC. This number resulted from 978 bytes payload, 12 bytes for the RTP header (we use the same size for the TFMCC header), 8 bytes for the UDP header and 20 bytes for the IP header. We measure the fairness of the two protocols by using Jain's Fairness Index [33], which is defined as follows:

$$f\left(x_1, x_{2,}, x_{3,}, \ldots, x_n\right) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2} \tag{7}$$

in which x represents the throughput of n flows that share network resources.

For this simulation we create a YUV video sequence that consists of 10 000 frames with a duration of 400 seconds. Figures 3 and 4 present the results of separate ASMP and TFMCC simulations. The transmission rate for both TFMCC and ASMP is adjusted based on the feedback reports from the slowest receiver in a session. For easier observation, we present only the results from one multicast receiver in both cases (ASMP and TFMCC) because all of them are behind the same bottleneck link and receive the same multimedia flow.

TFMCC has higher performance in terms of the achievable throughput than ASMP at the expense of higher packet losses. The ASMP packet loss ratio is measured to be 0.16%, while TFMCC presents a packet loss ratio of 0.67%. The packet loss ratio has a negative impact on the video quality as seen by the end user and we will observe this negative impact on the PSNR values later in this paper.

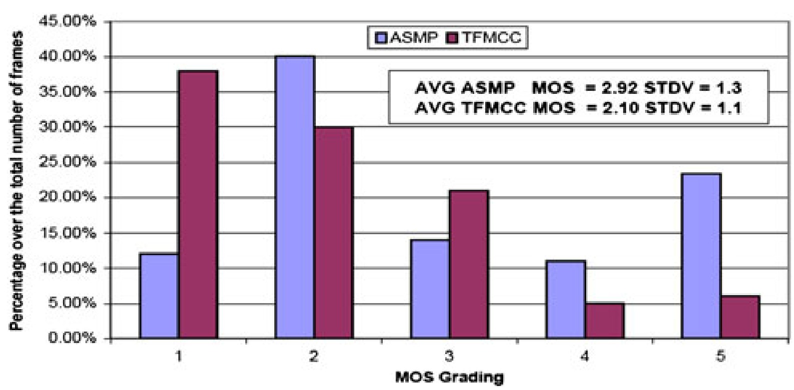

Figure 6. Resulting MOS values of TFMCC and ASMP

DOI: 10.1002/nem

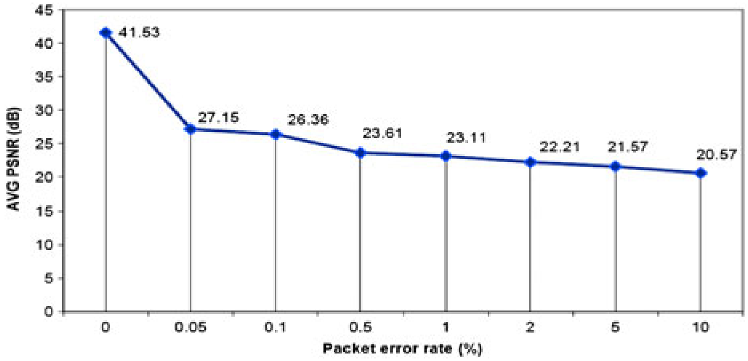

Figure 7. Average PSNR as a function of packet loss ratio

Jain’s fairness index is higher for TFMCC, which indicates that the available bandwidth is better distributed between the TCP and the multicast TFMCC traffic (1.0 is perfect fairness). On the other hand, ASMP is more TCP-friendly than it ought to be and this is a direct result of its smoothing functions, which reduce responsiveness to rapid network changes. Another observation from these results is that both protocols present high TCP-friendly behavior as they almost equally share the bandwidth in the bottleneck link with TCP traffic.

At this point, we need to mention that TFMCC seems to have better performance than ASMP when taking into account network-centric metrics (throughput and Jain’s fairness index). However, as we will present later in the paper, ASMP provides to the end user better video experience when measuring video-centric metrics (PSNR and MOS). The reason behind this is that under the same network conditions ASMP presents lower packet losses than TFMCC. This is an important indication that evaluation of multimedia transmission strategies based only on network metrics can lead to debatable results in respect of the video quality that is finally offered to the end user.

6.2. *Video objective performance metrics*

Following our evaluation we measure the PSNR and MOS values. To obtain the PSNR values (Figure 5) we compare the encoded video at the sender with the video received by the receivers. The results show that ASMP clearly outperforms TFMCC, although a visual observation suggests that they present similar performance.

PSNR values are matched to MOS values (Figure 6) to obtain subjective evaluation results. We observe from the MOS values that ASMP video grading is mainly in the range ‘poor’ to ‘excellent’ and almost 50%

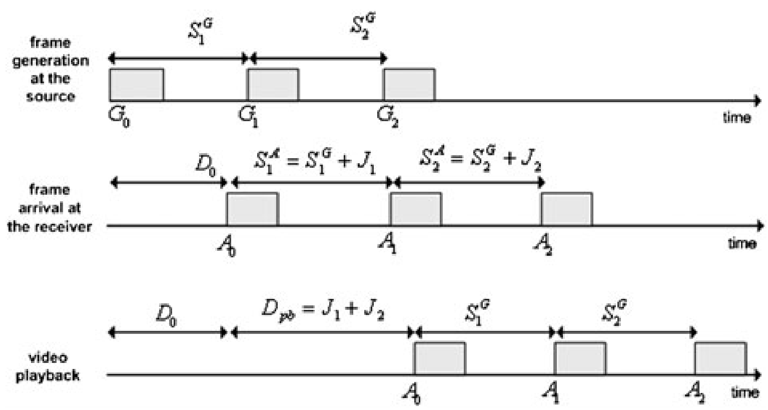

Figure 8. Media synchronization timing diagram

DOI: 10.1002/nem

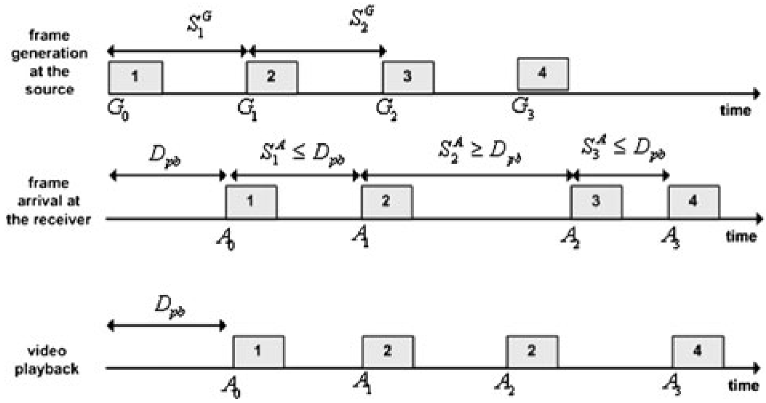

Figure 9. Effects of absolute play-out time constraint

of the video sequence is graded 'fair' to 'excellent'. TFMCC's performance is in the range 'bad' to 'poor' for the largest part of the video sequence, with only a small part that is graded 'fair' to 'excellent', which reflects approximately 30% of the entire video file. The PSNR degradation in the TFMCC case is linked to the higher packet loss ratio (0.67%) due to the congestion in the bottleneck link. The congestion avoidance algorithm of ASMP minimizes packet losses and increases the video quality, which is the most important attribute of any congestion control mechanism for multimedia data transmission.

6.3. *Effects of packet losses on video quality*

During our simulations we observed that even a small number of lost packets have a significant impact on PSNR values. This observation raised the question of how loss tolerant the multimedia applications are, as even a small number of lost packets degrade significantly the quality of the received video. Driven by our observations we conducted a number of simulations in order to investigate the effects of packet losses on the video quality due to congested or lossy links. Under the same network topology in Figure 2, and by excluding the TCP background traffic, we implemented a periodic error model and set the packet error rate from 0.05% to 10%. Figure 7 depicts the PSNR values in respect of the packet error rates. The results are similar for both protocols.

We observe that the implications on video quality become obvious with even a small number of lost packets. PSNR drops by approximately 14 dB in the case of a packet error rate of 0.05%. Higher packet losses lead to low PSNR values and as a result to low performance in terms of video quality. Therefore, it is important for a video application to engage additional error resilience methods in addition to congestion control mechanisms in order to minimize the effects of packet losses.

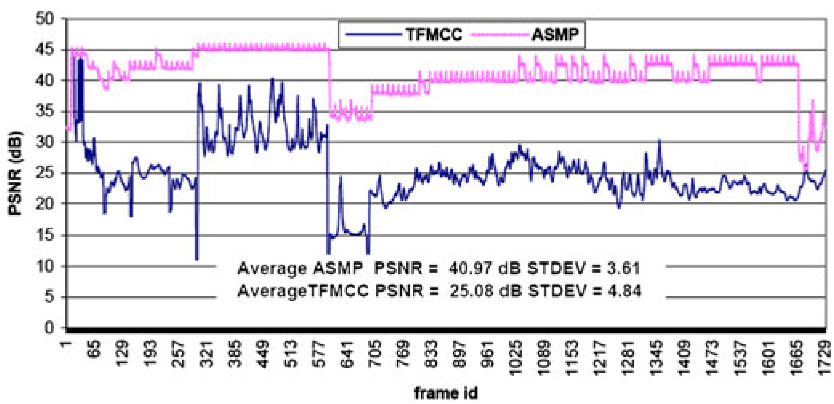

Figure 10. Resulting PSNR values (frame by frame) without absolute play-out buffer time constraint

DOI: 10.1002/nem

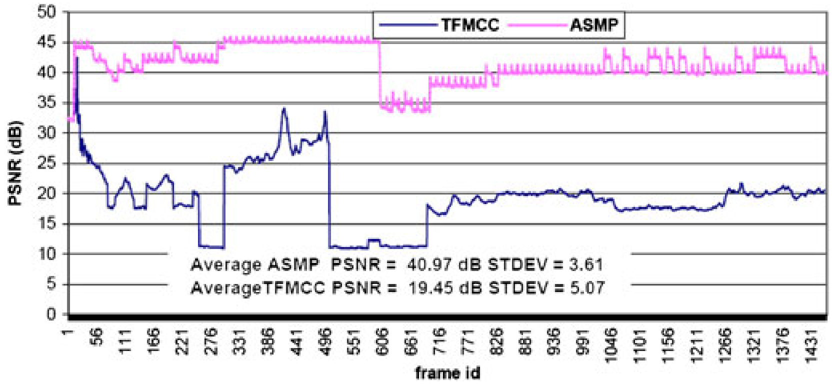

Figure 11. Resulting PSNR values (frame by frame) with absolute play-out buffer time constraint (150 ms)

At this point, we need to clarify that error control is beyond the scope in this paper. However, in our future work we plan to investigate an error control module for the ASMP protocol based either on FEC (forward error correction) or on selective retransmission of lost packets.

6.4. *The effects of cumulative jitter*

What is also important is the effect of cumulative jitter delay when dealing with video streaming applications. The cumulative jitter is defined as the amount of playback delay that must be provided in order to avoid discarding delayed video frames at the client side. Figure 8 depicts the frame generation at the sender as well as their arrival and playback at the receiver. For two consecutive frames $I-1$ and i the jitter delay variation is defined as follows:

$$J_i = S_i^A - S_i^G = (A_i - A_{i-1}) - (G_i - G_{i-1}) \tag{8}$$

The cumulative inter-frame jitter or simply cumulative jitter is defined [22] as ‘the amount of playback delay D_{pb} that must be provided to avoid discarding any frame in the sequence’:

$$CJ_k = \sum_{i=1}^{i} J_i \tag{9}$$

where (Figure 8) D_0 stands for the propagation delay, G_i is the sending time of frame i, S_i^G is the time spacing between two consecutive video frames $(I-1)$ and i, A_i is the arrival time of frame i, and S_i^A is the time spacing between two arriving video frames at the receiver. This metric is very important for assessing the performance of the underlying congestion control mechanism because it is highly correlated with the level

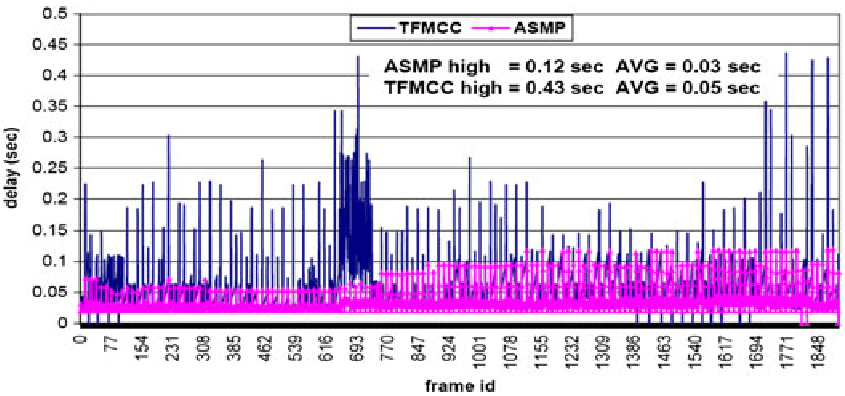

Figure 12. TFMCC and ASMP end-to-end delay measurements

DOI: 10.1002/nem

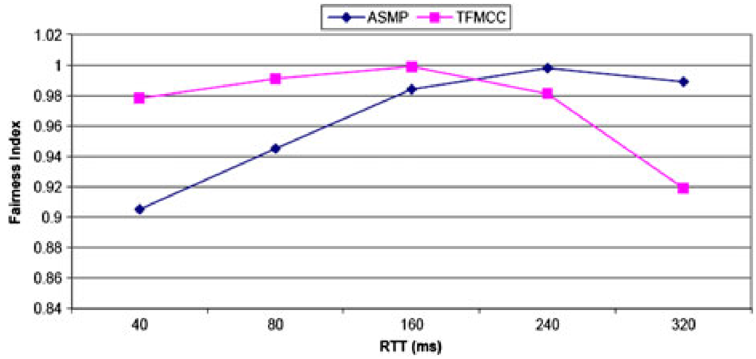

Figure 13. Jain's fairness index

of congestion in the bottleneck link(s). Once the cumulative jitter of a video frame exceeds the playback buffer duration D_{pb}, the video frame is useless and is discarded. A larger playback buffer allows late-arriving frames to be recovered at the expense of longer start-up time and extra storage capacity at the end user. We should highlight that a large playback buffer size can isolate network delivery delays but leads to important playback delays that may not be acceptable for interactive multimedia applications such as videoconferencing. In interactive multimedia applications important delays may cause important problems to the application usage.

6.5. *Simulations with delay constraints*

In order to assess the video quality at the end user when delay constraints are involved we use a dumbbell topology without any background traffic and set the one-way delay to 20 ms. Under this scenario we specify an absolute play-out buffer time relative to the frame transmission time, due to the real-time constraint. In this case video frames that are received with a delay less than or equal to the absolute play-out buffer D_{pb} will be rendered at the correct sequence position. Video frames that are received with delay larger than D_{pb} cause a previous frame to be displayed when that frame had to be displayed. The latest-arriving frame is not dropped, however, since it is the most recently received video frame. To explain better the consequences of the absolute play-out buffer time let us consider the example in Figure 9. Frame 2 is displayed twice due to increasing network delay. Frame 3 is never displayed. When it was received, frame 4 had also been received, which was more up to date according to the planned schedule.

In our simulations we set the absolute play-out buffer time at 150 ms, which reflects the recommended one-way delay for conversational media. We create a raw video that consists of different video sequences differing in complexity in the following order: *News* (frames 1–300) has medium complexity; *Akiyo* (300–600) has very low complexity, as it contains more static information; *Stefan* (600–700) is very complex due to continuous moving pictures; and lastly *Paris* (700–1753) has high to medium complexity. This sequence is encoded in MPEG-4 format with a temporal resolution of 25 frames per second, and GOP

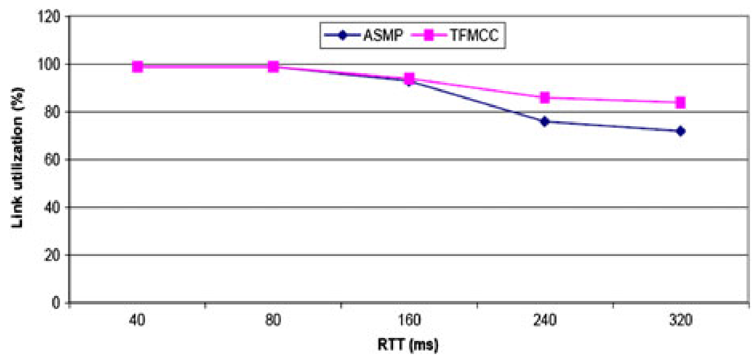

Figure 14. Link utilization

DOI: 10.1002/nem

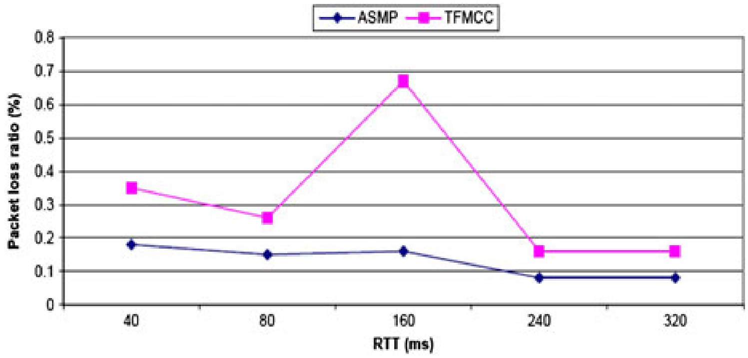

Figure 15. Packet loss ratio measurements

with a size of 12 frames per GOP. By combining various video sequences of different complexity we can better simulate video transmission as the temporal resolution changes over time.

Figure 10 depicts the PSNR values without any play-out buffer time constraint. This means that all received packets are used at the decoder in the frame assembly process. ASMP presents very high performance with PSNR values above 30 dB, while TFMCC does not scale well with the changing complexity of the video file over time. We observe that with the *Akiyo* video sequence (frames 300–600) PSNR values for TFMCC are above 30 dB. With the *Stefan* sequence (frames 600–700) PSNR values drop by approximately 15 dB as the video complexity increases.

ASMP scales better even with video sequences of high temporal resolution. We observe that the *Stefan* sequence causes a reduction to PSNR values by approximately 10 dB. Clearly, in this simulation scenario ASMP outperforms TFMCC.

Next, in Figure 11 we observe that the play-out constraint of 150 ms affects mainly TFMCC's performance. PSNR values for TFMCC drop by approximately 5 dB in all video sequences. This is a direct result of higher end-to-end delay values. TFMCC presents high one-way delay, while ASMP's one-way delay is lower than 150 ms (Figure 12); at equilibrium, it is below 50 ms. Our observation is that ASMP's performance is not affected by the 150 ms play-out constraint.

6.6. Simulations with varying RTT values

To obtain a better understanding of the dependencies between the video quality, achieved throughput and packet loss ratio we run several simulations with different RTT values on the path between the video sender and the multicast receivers. We use the same topology in Figure 2 with TCP background traffic.

TFMCC presents higher fairness than ASMP in the case of RTT values below 160 ms. We note that as the RTT increases TFMCC occupies a higher portion of the available bandwidth in the bottleneck link than TCP. TCP seems to suffer when it shares network resources with TFMCC in cases with RTT values above 200 ms (Figure 13). On the other hand, ASMP retains its TCP-friendly behavior in all simulation scenarios. The reason behind this is that TFMCC is more aggressive than ASMP, which causes TCP to frequently back off. This becomes more obvious in Figure 14, when we plot the achieved link utilization of the two protocols over different RTT values. TFMCC outperforms ASMP in all cases and is a better solution when bandwidth utilization is a major concern. However, the higher bandwidth utilization leads to higher packet losses (Figure 15) and introduces larger oscillations. To assess the stability of the two protocols, in terms of minimizing oscillations of throughput, we use the coefficients of variation (CoV)[3] of the throughput values and plot the results in Figure 16. We observe that ASMP presents better stability than TFMCC.

It is also interesting to note that the low throughput values provide better video quality as the packet loss ratio is also low. This is the case in all simulation results when we directly compare the achieved

[3]*Coefficient of variation (CoV) is the standard deviation divided by the mean.*

DOI: 10.1002/nem

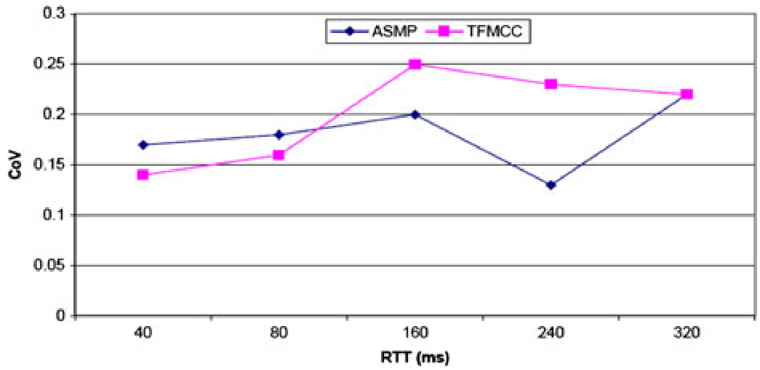

Figure 16. CoV over different RTTs

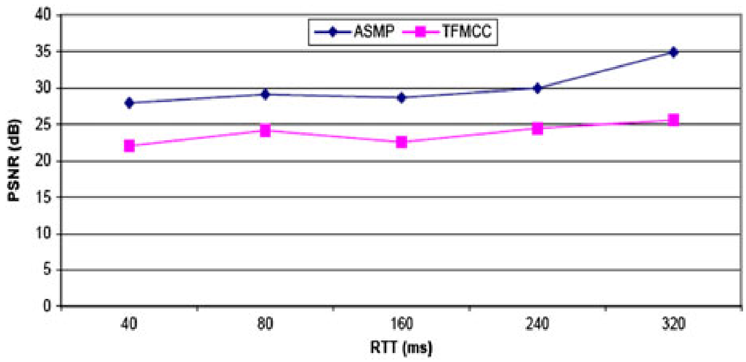

Figure 17. Video quality

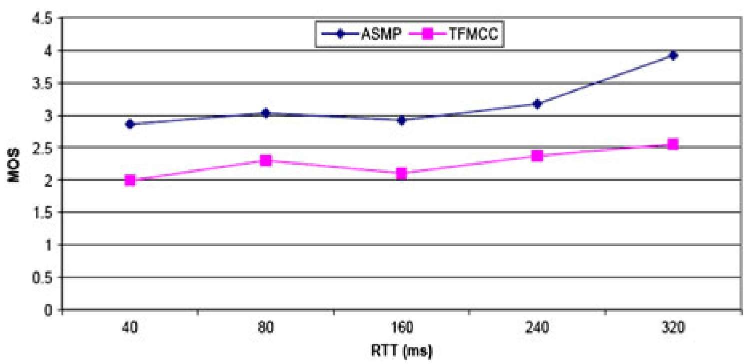

Figure 18. MOS grading

video quality of the two protocols (Figure 17). In all cases TFMCC presents lower PSNR values (approximately 5 dB) than ASMP. Our assessment is that the aggressiveness of any transport protocol for multimedia applications should be carefully defined as it may lead to frequent packet losses that undermine the video quality. The aggressiveness is defined in RFC 5166 as 'the maximum increase in the sending rate in one RTT, in packets per second, in the absence of congestion'. TFMCC is more aggressive than ASMP in order to increase throughput, although both TFMCC and ASMP use almost the same TCP equation to calculate a TCP-friendly share when they detect packet losses. The difference is that ASMP's maximum increase is bounded between 0.9 packet/s and 0.2 packets/s, in accordance with equation (5). We argue that by early detection of the upcoming congestion and 'slowing down' the constant increase of 1 packet/s to a lower rate we can avoid unnecessary packet losses.

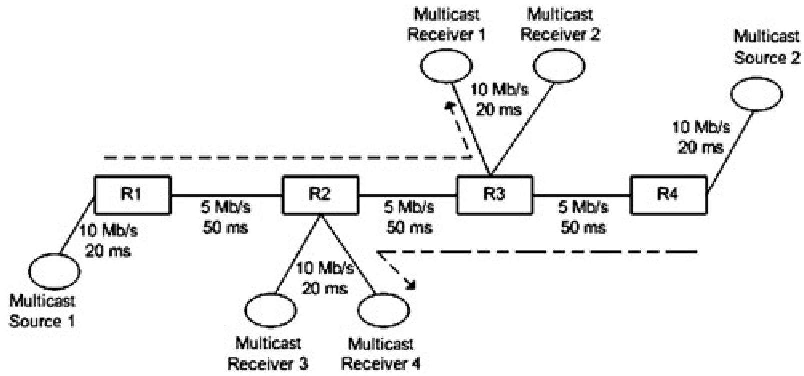

Figure 19. Network topology for intra-fairness evaluation

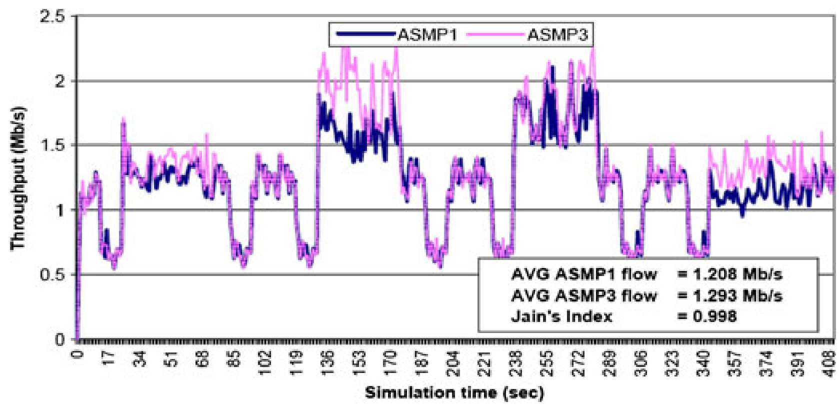

Figure 20. ASMP intra-protocol achieved throughput

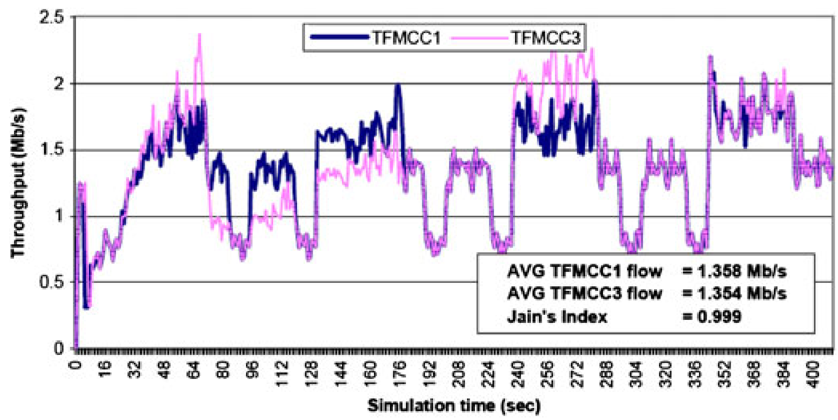

Figure 21. TFMCC intra-protocol achieved throughput

When TFMCC detects packet losses the network is already congested. MOS grading discloses that ASMP clearly outperforms TFMCC in all simulations with different RTT values (Figure 18).

Therefore, moderate and stable transmission rates with minimum losses provide a better service to the end user in terms of video quality. Otherwise, error resilience mechanisms should be applied to avoid at least the losses of I-frames that have the highest importance in the video sequence. ASMP performs better in topologies with realistic RTT values of several hundreds of milliseconds as the protocol's smooth functions prevent high oscillations, which definitely lead to packet losses.

DOI: 10.1002/nem

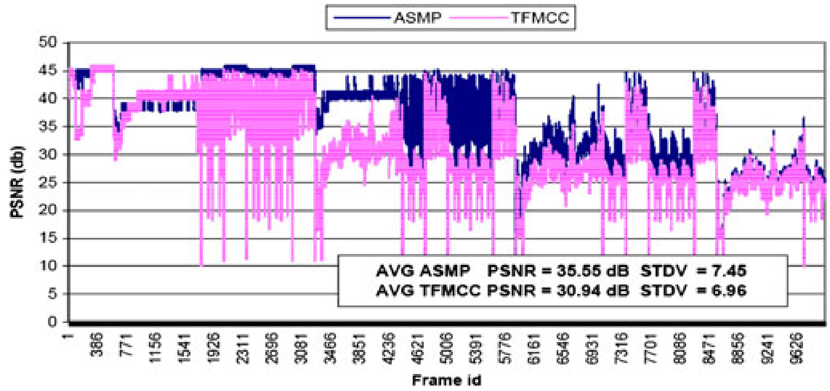

Figure 22. Resulting PSNR values (frame by frame) of TFMCC and ASMP

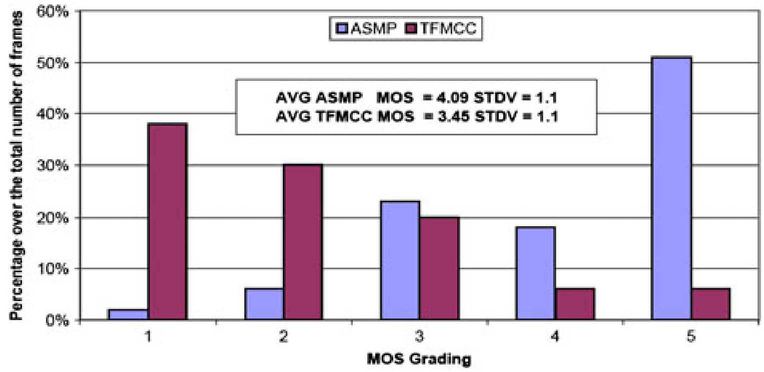

Figure 23. Video grading

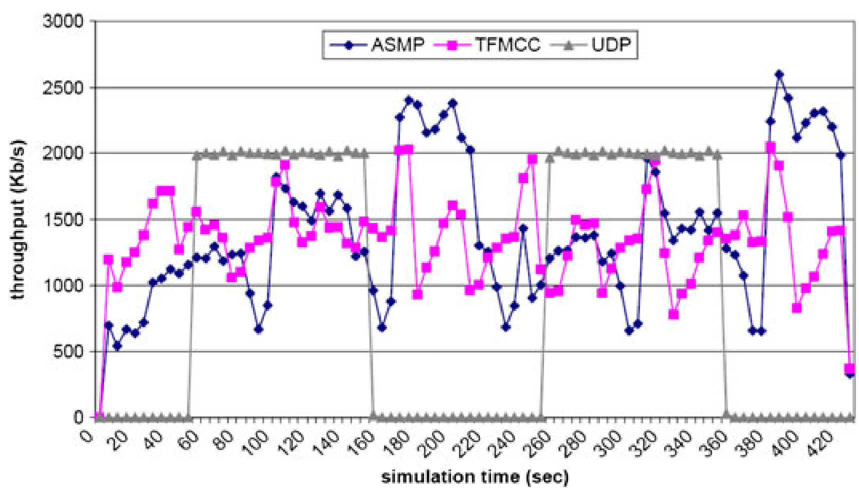

Figure 24. Throughput of TFMCC and ASMP with competing UDP flow

6.7. *Intra-protocol fairness*

In this simulation scenario, we evaluate the intra-protocol fairness by connecting two video sources using the same protocol (ASMP or TFMCC) which transmit the same video file, with a duration of 408 seconds (6.8 minutes), to two different multicast groups as shown in Figure 19. The two video sources transmit to opposite directions via links with the same bandwidth and propagation delay

DOI: 10.1002/nem

and share the bottleneck link between R2 and R3. We use Drop Tail queues in routers R1 to R4 and set the same packet size for the two video sources. For easier observation we present the simulation results from two representative receivers of each group (Figures 20 and 21) as all the receivers in each case (ASMP or TFMCC simulations) are behind the same bottleneck link and receive the same multimedia flow). The results from the different simulation sets disclose that both TFMCC and ASMP sources fairly share the available bandwidth in the bottleneck link, with a measured Jain's fairness index of 0.99. However, TFMCC presents higher link utilization than ASMP, as has been observed already in all of our simulations at the expense of higher packet losses. This becomes obvious when we compare the achieved PSNR values (Figure 22), which show that on average ASMP outperforms TFMCC by 5 dB. We can observe the high diversity of PSNR values with TFMCC, which is again a direct result of packet losses due to higher congestion in the bottleneck link. MOS grading (Figure 23) shows that in the TFMCC case approximately 70% of the received video file is graded between 'unacceptable' and 'poor'. ASMP presents very high performance, as 50% of the received video is graded as 'excellent', while 90% of the received video is graded between 'fair' and 'excellent'.

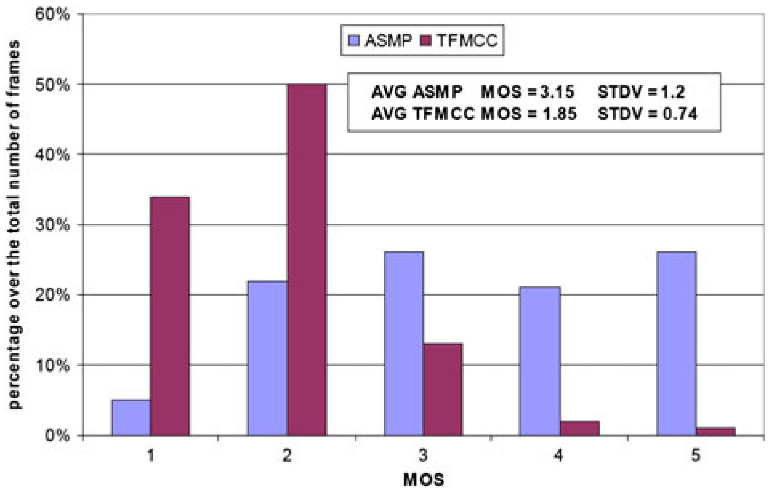

Figure 25. Video grading with UDP competing flow

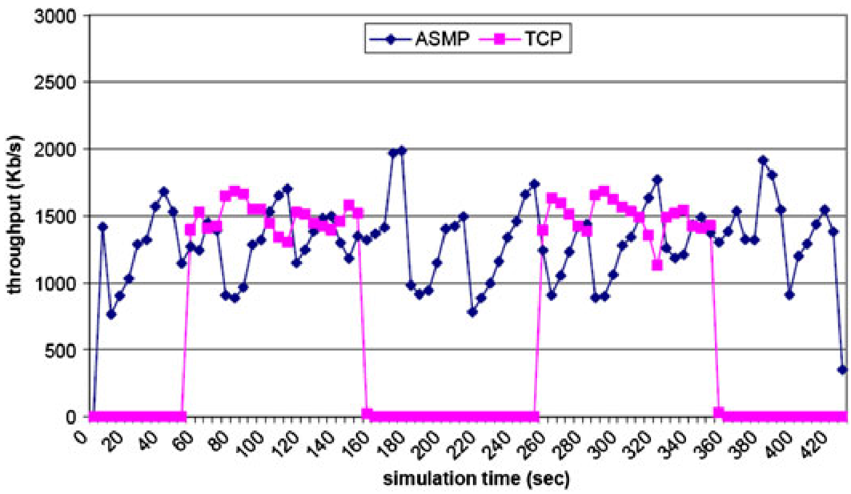

Figure 26. Throughput of ASMP and competing TCP flow

DOI: 10.1002/nem

6.8. *Responsiveness to dynamics of competing UDP and TCP traffic*

In this simulation we investigate the ability of ASMP and TFMCC to react and adjust the sender's transmission rate when competing for network resources with TCP and UDP traffic which does not employ any congestion control mechanism. We use a dumbbell simulation scenario with UDP background traffic. The bottleneck link has 4 Mb/s capacity and 10 ms delay. To better test the responsiveness of the two protocols we vary the available bandwidth in the bottleneck link as a square wave by injecting UDP traffic throughout the simulation time. UDP traffic is transmitted by a constant bit rate (CBR) source at 2 Mb/s, which occupies one half of the available bandwidth.

TFMCC presents higher throughput than ASMP and reacts faster to network changes due to UDP traffic. However, TFMCC overshoots, as in the slow start phase the sending rate is increased exponentially unless a packet loss event is observed. We can observe the increase of the transmission rate of TFMCC in the first simulation seconds (Figure 24), in which the transmission rate is higher than that of ASMP. The direct result is higher packet loss ratio values (0.92%). ASMP is a conservative congestion control mechanism at the expense of lower bandwidth utilization. It does not employ any slow start

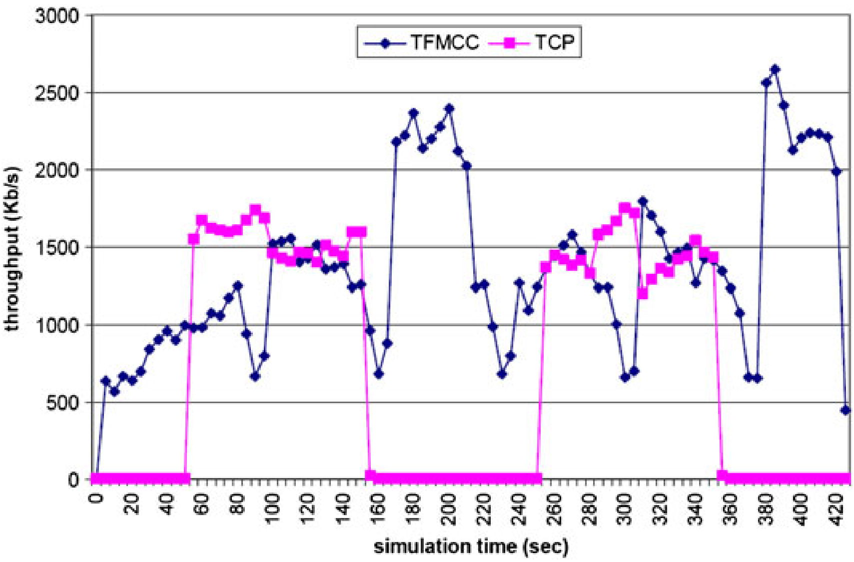

Figure 27. Throughput of TFMCC and competing TCP flow

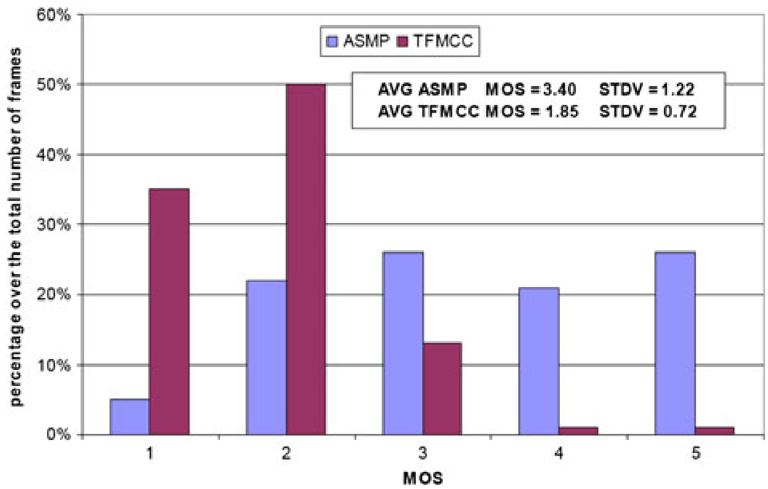

Figure 28. Video grading with TCP competing flow

DOI: 10.1002/nem

mechanism and increases its transmission rate gradually by almost 0.9 packets per RTT. It takes longer for ASMP to reach its highest transmission rate than TFMCC but it ensures a smooth increase of the transmission rate in order to avoid packet drops. The packet loss rate in ASMP is very low (0.046%). MOS values of the received video are presented in Figure 25. We observe that ASMP clearly outperforms TFMCC as approximately 75% of the received video is graded from 'fair' to 'excellent'.

In the second simulation scenario we inject TCP background traffic in a similar way to the previous simulation. We observe in Figures 26 and 27 that ASMP presents again a smoother behavior than TFMCC. The packet loss ratio values are similar to the previous simulation with UDP competing traffic (0.037% for ASMP and 0.92% for TFMCC), resulting in the MOS values depicted in Figure 28. Once again we observe that ASMP outperforms TFMCC. Although the above simulation scenario is very simple, it does provide clear indications that TFMCC's high responsiveness increases link utilization at the expense of lower video quality at the end user.

7. CONCLUSIONS

We presented in this work the lessons learned from the performance evaluation of adaptive congestion control schemes for MPEG-4 video transmission. We used an integrated simulation environment that combines ns-2 simulation software with Rate Adaptive Evalvid (Evalvid-RA). Additional codes were added to extend Evalvid-RA to the multicast domain in order to exploit the feedback functions of the well-accepted RTP/RTCP protocols. The RTCP sender and receiver reports eliminated the need for additional feedback mechanisms. Simulation results under that new simulation environment came closer to a real video experimental evaluation process.

The transmitted and received video files were directly compared in order to provide PSNR values that indicated the quality of the received video. We mapped the PSNR values to MOS grades based on ITU-R recommendations. However, it should be pointed out that this mapping provided only a rough estimation of user perception. When possible, experiments with real human interactions would provide more precise and realistic results.

Both protocols proved to be TCP-friendly as they fairly shared the available bandwidth with TCP flows. The TCP-friendliness is defined differently in ASMP from that in TFMCC, in which the sending rate is generally within a factor of two of the sending rate of a TCP flow that traverses the same link with that TFMCC flow. ASMP is a 'moderate' protocol in which the sending rate consumes no more bandwidth than a TCP connection that is traversing the same path as ASMP.

TFMCC proved to be a more efficient congestion control mechanism than ASMP when network utilization was of major concern. However, the high transmission rates of TFMCC tended to rapidly occupy the largest portion of the available bandwidth in the bottleneck link, which created congestion and frequent packet losses. The direct result of those packet losses was PSNR degradation. ASMP proved to be a more efficient multicast transmission mechanism than TFMCC when end user experience was of major concern. The smooth operations of ASMP led to fewer packet losses and better quality of video transmission (in terms of PSNR and MoS). In addition, ASMP presented high responsiveness to packet losses and adapted rapidly to changes in the network, although the convergence time was higher than that of TFMCC.

Packet losses had a negative effect on the video quality as even a small number of lost packets degraded the achieved PSNR. Therefore, moderate transmission rates that satisfy the video application requirements are preferable to avoid packet losses. Otherwise, error resilience mechanisms should be applied to recover from losses, especially from I-frames that cannot be recovered by the video decoders at the end user.

The aggressiveness of a transport protocol should be balanced between achieved throughput, packet losses and stability as high aggressiveness led to oscillations of the transmission rate. Those oscillations in the transmission rate had an undesired effect in video transmission. ASMP demonstrated its ability to deliver MPEG-4 video files with high quality. The small cumulative jitter values reduced large playback delays and increased QoE.

Simulation results with competing UDP and TCP traffic disclosed that ASMP was able to adjust and reduce the transmission rate to avoid packet losses. Therefore, the responsiveness of ASMP was assessed

DOI: 10.1002/nem

as satisfactory when taking into account the rather large feedback timeout of the RTCP protocol. TFMCC had faster reactions than ASMP at the expense of lower video quality.

Uncontrolled video transmission without any flow/congestion control mechanisms should be avoided as it leads to poor-quality service. However, congestion control and avoidance mechanisms by themselves are not the panacea to solve all issues related to video transmission. More efficient video encoding techniques along with error resilience and higher access capacity are the key elements for increasing video quality. Moreover, comparison of ASMP against TFMCC has shown that the performance evaluation of such protocols should not only be based on network-centric metrics. Application-based quality metrics should also be taken into account. This becomes obvious from the fact that, although the comparison of ASMP with TFMCC has shown that TFMCC has better performance in terms of bandwidth utilization and responsiveness, ASMP presented higher performance in terms of video quality.

8. FUTURE WORK

Apart from the observations made during the simulations in respect of ASMP behavior, we believe that further studies on the '*smooth*' transmission rate concept will benefit research in the area of multicast transmission. We still need to investigate ASMP's scalability with a larger number of receivers (e.g. thousands of receivers). Moreover, we plan to investigate in the future how to enhance our implementation by adding a mechanism in order to dynamically choose and modify the parameters that regulate the aggressiveness of ASMP.

Congestion Indicators that are based on statistics of jitter delay measurements are very well suited to wireless networks in which delay measurements exploit more effectively and more accurately the network congestion level than packet loss events.

Moreover, we will investigate more deeply the effect of '*smoothening the transmission rate*' on other competing traffic types and loss error schemes. It is also our intention to use our solution as part of a multi-rate transmission scheme.

In addition, we plan to implement a prototype of ASMP in order to evaluate its performance in real networks. We also plan to investigate an error control module for ASMP based either on FEC (forward error correction) or on selective retransmission of lost packets.

Finally, the source code of ASMP implementation with Multi-Evalvid-RA support and documentation are available [31].

REFERENCES

1. Wang B, Kurose J, Shenoy P, Towsley D. Multimedia streaming via TCP: an analytic performance study. *ACM Transactions on Multimedia Computing, Communications and Applications* 2008; **4**(2): Article 16.
2. Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: a transport protocol for real-time applications. *RFC 3550*, July 2003.
3. Widmer J, Handley M. On TCP-friendly multicast congestion control (TFMCC). *RFC 4654*, August 2006.
4. Bouras C, Gkamas A, Kioumourtzis G. Evaluation of single rate multicast congestion control schemes for MPEG-4 video transmission. In *5th EURO–NGI Conference on Next Generation Internet Networks (NGI 2009)*, Aveiro, Portugal, 1–3 July 2009.
5. The UCB/LBNL network simulator, software. Available: http://www.isi.edu/nsnam/ns/ [3 December 2011].
6. Widmer J, Handley M. Extending equation-based congestion control to multicast applications. In *Proceedings of ACM SIGCOMM '01*, 2001.
7. Handley M, Floyd S, Padhye J, Widmer J. TCP friendly rate control (TFRC), Network Working Group. *RFC 3448*, January 2003.
8. Rizzo L. *pgmcc*: a TCP-friendly single-rate multicast congestion control scheme. In *Proceedings of ACM SIGCOMM '00*, 2000.
9. Smith H, Mutka M, Rover D. A feedback based rate control algorithm for multicast transmitted video conferencing. *Journal of High Speed Networks* 1998; **7**: 259–279.
10. Sisalem D, Wolisz A. LDA + TCP-friendly adaptation: a measurement and comparison study. In *Proceedings the 10th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2000)*, Chapel Hill, NC, 25–28 June 2000.
11. Macker J, Adamson R. A TCP friendly, rate-based mechanism for nack-oriented reliable multicast congestion control. In *Proceedings of IEEE GLOBECOM*, 2001.
12. Jiang L, Yuksel M, Kalyanaraman S. Explicit rate multicast congestion control. *Computer Networks* 2006; **50**(15): 2614–2640.

DOI: 10.1002/nem

13. Xie F, Feng G. Chee Kheong Siew. Study on nominee selection for multicast congestion control. *Computer Communications* 2006; **29**(9): 1458–1469.
14. Chen H, Liu Z, Chen Z, Yuan Z. Extending TCP congestion control to multicast. *Computer Networks* 2007; **51**(11): 3090–3109.
15. Ma H, Meng X, Zhou L, Li H, Zhang X. Fuzzy-logic-based rate adaption scheme for TFMCC. In *International Conference on Networking and Digital Society*, Vol. **1**, 2009; 238–241.
16. Kammoun W, Youssef H. Equation-based end-to-end single-rate multicast congestion control. *Annals of Telecommunications* 2010; **65**(3–4): 219–231.
17. *ITU-T Recommendation G.100/P.10*. Amendment 1: New appendix I—Definition of quality of experience (QoE), January 2007.
18. 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, IL, 2003.
19. Lie A, Klaue J. Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR video. *Multimedia Systems* 2007; **14**(1): 33–50.
20. Bouras C, Gkamas A, Kioumourtzis G. Extending the functionality of RTP/RTCP implementation in network simulator (ns-2). In *First International Conference on Simulation Tools and Techniques for Communications, Networks, and Systems*, Marseille, France, 3–7 March 2008.
21. Pandhye J, Kurose J, Towsley D, Koodli R. 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.
22. Tionardi L, Hartanto F. *The use of cumulative inter-frame for adapting video transmission rate*. In *Proceedings of TENCON*, Bangalore, India, 2003.
23. Bouras C, Gkamas A, Kioumourtzis G. Adaptive smooth multicast protocol for multimedia transmission: implementation details and performance evaluation. *International Journal of Communication Systems* 2010; **23**(3): 299–333.
24. *ITU-T Recommendation P.910*. Subjective video quality assessment methods for multimedia applications, April 2008.
25. Rohaly M, Corriveau P, Libert J, Webster A, Baroncini V, Beerends J, Blin J-L. Video Quality Experts Group: current results and future directions. In *SPIE Visual Communications and Image Processing*, Vol. **4067**, Perth, Australia, 21–23 June 2000; 742–753.
26. *ITU-T Recommendation G.1080*. Quality of experience requirements for IPTV services, December 2008.
27. Winkler S. *Digital Video Quality: Vision Models and Metrics*. Wiley: Chichester, 2005.
28. FFmpeg audio/video conversion tool. Available: http://sourceforge.net/projects/ffmpeg/ [3 December 2011].
29. *ISO/IEC 14496-2*. Information technology – Coding of audio-visual objects – Part 2: Visual, 1999.
30. EvalVid - A Video Quality Evaluation Tool-set. Available: http://www.tkn.tu-berlin.de/menue/research/evalvid/
31. ns-2 Contributed Code. Available: http://nsnam.isi.edu/nsnam/index.php/Contributed_Code#Transport [3 December 2011].
32. Floyd S. Metrics for the evaluation of congestion control mechanisms. *RFC 5166*, Network Working Group, March 2008.
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*, September 1984.

AUTHORS' BIOGRAPHIES

Georgios Kioumourtzis obtained his Diploma from the Hellenic Army Military Academy and graduated as Signal Officer in 1986. In 1996, he extended his studies in the Telecommunications School for Signal Officers. He also received two Master Degrees, one in Computer Science and the second in Systems Engineering from the Naval Postgraduate School CA, USA in 2005. In 2010, he received his Ph.D from the Computer Engineering and Informatics Department of Patras University (Greece). His research interests include Computer Networks, Simulations, Multimedia transmission, Wireless Ad Hoc Networks and Cross Layer Design. He has published more than 20 research papers and journals in well-known refereed conferences.

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 coauthor 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 J Netw Syst Manage 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 Ph.D from the Computer Engineering and Informatics Department of Patras University (Greece). He is currently Assistant Professor in University Ecclesiastical Academy of Vella, Ioannina and visiting lecturer in Hellenic Open University in the Informatics Curriculum. 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, and Information Society.