Content-aware Interface Selection Method for Multi-Path Video Streaming in Best-effort Networks

Árpád Huszák, Sándor Imre
Budapest University of Technology and Economics, Department of Telecommunications
H-1117 Budapest, Hungary
{huszak, imre}@hit.bme.hu

Abstract— **Today's mobile devices are equipped with multiple interfaces to make the connection possible to different type of networks. In order to efficiently utilize the interface capabilities, multi-path streaming can be used. Resource intensive applications can deliver high bitrate streams over multiple paths by cumulating the available bandwidth of the different subpaths. In this paper we propose a multi-path streaming method that chooses a set of paths maximizing the overall quality at the client. While the available paths have different bandwidth, delay and loss probability constrains, the packet distributor must take the video packet importance and the dependencies between packets into account. Transmitting the reference video frames on the most reliable links will decrease the loss probability of important data packets and increase the measured video quality. Both analytical and simulation analysis were performed to examine the behavior of the presented content-aware interface selection model. The results show that the proposed solution provides higher video quality than common scheduling algorithms.**

Index Terms—**video streaming, multi-path delivery, Quality of Service**

I. INTRODUCTION

Multimedia applications are gaining prominence on the Internet in both wired and wireless environment. Despite the development of new access technologies that provide higher bandwidth to the users, multimedia streaming applications still suffer from limited and highly varying bandwidth. However the offered bandwidth is getting higher, high quality video streams usually can not be transmitted on a single link. Multi-path video streaming has recently been proposed as a solution to overcome the strict mobile network limitations. In order to increase the quality of streams using multihomed devices that have multiple network interfaces independently connected to different networks, all the available interfaces must be effectively utilized. Higher bandwidth can be provided to the users if joint links are used. The independent but joint links can be successfully utilized by a streaming application in order to achieve high quality streaming.

Multi-path multimedia streaming has become attractive solution due to the increased throughput and adaptive load balancing [1]. An overview of the benefits and use of path diversity for media streaming is introduced in [2][3]. In these work different approaches for media coding and streaming over multiple paths are examined, together with architectures for achieving path diversity between single or multiple senders and a single receiver. Novel solutions were also presented to improve the quality of streaming applications. The authors of [4][5] study video streaming using feedback-based reference picture selection schemes and path diversity. In [6] multi-path streaming methods are examined from the application transparency point of view. A distributed sender-driven streaming solution is presented in [7], where servers collaboratively adapt to the network status in order to provide the media client with a superior video quality. Most of the related works analyze the multi-path streaming solution using simulations; however the authors of [8] deployed a testbed for the measurements. The multi-path streaming was also investigated from congestion point of view in [9][10]. In the latter one a fast heuristic-based algorithm is presented, built on load-balancing principles.

Multihomed protocols such as SCTP [11] provides the possibility of multiple link connections, however the current protocol version does not allow to use multiple path simultaneously. Ye et al. [12] introduced an independent per path congestion control principle for standard SCTP, while in [13] a similar solution is presented with load sharing techniques.

In this paper an interface selection algorithm for MPEG video streaming is presented. The loss of packet in a video frames leads not only to reduced quality of the video but also results the propagation of distortion to successive frames. This is due to inter-frame-video compression algorithms such as MPEG, which exploit temporal correlation between frames to achieve higher compression. The proposed scheme takes the video packet importance and the dependencies between packets into account and transmits the reference video frames on the most reliable links. Using the proposed packet distributor the loss probability of important data packets can be decreased, hereby increasing the measured video quality. The main difference from the other multi-path streaming solutions is that we use a sender side buffer to heap up more data in order to utilize the available links more efficiently. We have analyzed the proposed scheme both analytically and with simulations to estimate the quality improvement of the transmitted video streams.

The rest of the paper is organized as follows. In Section II we introduce our interface selection method for MPEG video streams. The obtained performance results are presented in Section III. Finally, we summarize our paper and make the conclusions in the last section.

II. INTERFACE SELECTION SCHEME

We consider a client-server network topology, where the client can access the server simultaneously through different paths. The independent paths are characterized with different bandwidth (bw_i), packet loss ratio (p_i) and network delay (d_i) parameters belonging to path i.
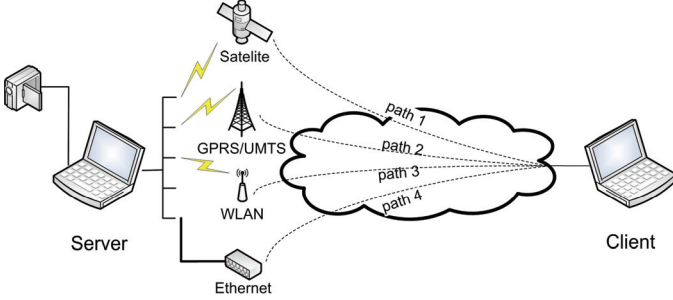

Fig. 1. Multi-path data delivery

Network delay parameter plays role when a strict playout buffer is used at the receiver. If the transmitted packet is received after the playout, it can be considered as lost packet, therefore we do not deal with the delay parameter separately. We consider that total packet loss probability (π_i) on path i contains the loss events due to channel errors and late reception of video packets:

$$\pi_i = p_i + \Pr(d_i > d_{pbuff}), \qquad (1)$$

where d_{pbuff} is the playout buffer delay. The overall packet loss ratio for the multi-path transmission is the weighted average of the total packet loss probabilities

$$\pi = \frac{d_1}{d}\pi_1 + \frac{d_2}{d}\pi_2 + ... + \frac{d_n}{d}\pi_n = \sum_{i=1}^{n}\frac{d_i}{d}\pi_i \qquad (2)$$

Where d_i is the data amount transmitted on link the ith link and d is the total size of the delivered video stream.

The transmitted video quality can be increased if the total packet loss probability is minimized; therefore the available paths are used in order of the links quality (π_i). The lowest packet loss ratio paths must be utilized first and the worst channels must be used only if necessary.

Further video quality improvement can be achieved if the packet contain is taken into account. The MPEG coded video streams use different frame types (I-, P- and B-frames) with inter-frame dependencies. We propose a content-aware packet distributor scheme in order to minimize the distortion propagation in the stream by forwarding the I-frames on the most reliable links.

In our proposed scheme a buffer is applied that heaps up

B_f number of frames causing d_B extra delay. In every d_B time we distribute the stored data packets between the interfaces and the heap up procedure starts again. This method makes it possible to efficiently distribute the data packets of the video stream. If longer delay (d_B) is used, more I-frames can be forwarded to the most reliable link, but the overall delay (D) will be also higher.

$$D = d_B + \max_i(d_i) \qquad (3)$$

The acceptable delay threshold depends on the used multimedia application. In case of one-way live media applications it can be even few seconds, but if two-way interactive application is used the perceptual tolerance of the user is around 150-200ms. The limitations of two-way interactive applications are too strict for our proposed interface selection algorithm, but for pre-recorded and one-way live video streams this solution is acceptable from the delay point of view.

The streaming server is connected to the channels through buffer interfaces, which are modeled as FIFO queues. These queues are refilled at every d_B time from the heap up buffer. The interface buffers' sizes are determinated by the paths bandwidth and the heap up buffer delay (d_B). The interface buffers must be purged in d_B time to be able to receive the next amount of data. The server side interface selection scheme is illustrated in Fig. 2.
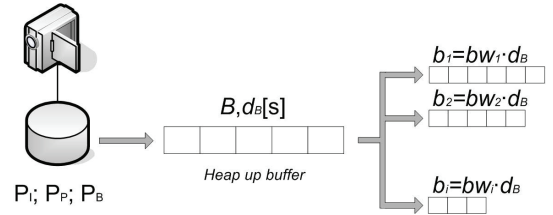

Fig. 2. Server side interface selection scheme

Quality degradation can be caused not only due to late packet delivery and channel errors, but also due to link overload. The bitrate variance of the coded video can be significant if VBR (variable bitrate) video codec is used. The needed bandwidth can be even higher than the joint links cumulated throughput capacity in a d_B length timeslot. Investigation results of Neame et al. [14][15]show that M/Pareto process can be used to accurately predict the queuing performance of a VBR broadband traffic stream. M/Pareto traffic is composed of a number of overlapping bursts. Bursts arrive according to a Poisson process with rate λ. The arrival process for each burst is constant for the duration of that burst, and has rate r. The mean amount of data (M) arriving within an interval of length t the M/Pareto traffic model is

$$M = \frac{\lambda \cdot t \cdot r \cdot \delta \cdot \gamma}{\gamma - 1}. \qquad (4)$$

Parameter γ stands for the minimum possible value of a Pareto random variable, while δ is the shape parameter, also known as the "tail index". Using (4), the load level of the heap up buffer can be estimated if a d_B long time period is examined. The probability of link overload (P_{OL}) is equal with possibility when the heap up buffer level is greater than the total amount of data transferable on the links within d_B time. Path i is characterized with its bandwidth (bw_i).

$$P_{OL} = \Pr\left(M > \sum_{i=1}^{n} bw_i \cdot d_B \right) \qquad (5)$$

The video quality improvement is gained due to the efficient packet distribution from the heap up buffer to the interface buffers. Due to the proposed content-aware interface selection method the distribution of data belonging to I-, P-, B-frames on the independent links will be totally different from the original video frame structure. On the reliable channels more I-frame data will be delivered and less B-frames, while on links with high loss probability mostly B-frames will be transmitted.
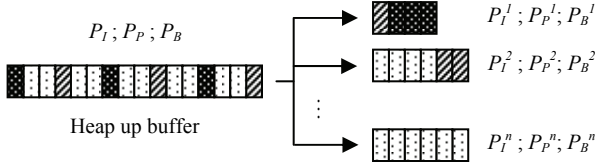

$$P_I ; P_P ; P_B$$

Heap up buffer

$$P_I^1 ; P_P^1 ; P_B^1$$
$$P_I^2 ; P_P^2 ; P_B^2$$
$$P_I^n ; P_P^n ; P_B^n$$

Fig. 3. Frame distribution on the interfaces

The possibility of the data bytes belonging to I-, P- and B-frames of the video stream is consequently P_I, P_P and P_B. If the heap up delay (d_B) is long enough these possibilities are similar in the heap up buffer. Due to the proposed interface selection algorithm the distribution of I-, P-, B-frame data bytes in the interface buffers will be different from the original video frame data distribution.

The possibility that a data byte belongs to an I-frame in the most reliable interface's buffer can be calculated as follows:

$$P_I^1 = \min\left(\frac{P_I B}{b_1}, 1 \right) \qquad (6)$$

Parameter B is the size of data heaped up, while B_1 is the buffer size of the first (most reliable) interface. We use the $min()$ function to keep the probability variable bellow or equal to 1.

In order to analyze the behavior of the proposed scheme we can calculate the distribution of I-, P-, B-frame data bytes in the interface buffers (P_I^i, P_P^i and P_B^i). The I-frame data probability in the ith interface buffer is:

$$P_I^i = \max\left(0, \min\left(\frac{P_I B - \sum_{n=1}^{i-1} b_n}{b_n}, 1 \right) \right) \qquad (7)$$

Similarly, the P- and B-frame data ratios in the ith interface buffer can be calculated as follows:

$$P_P^i = \max(0, \min((\max(0, \min(\frac{P_P B - \sum_{n=1}^{i} P_P^n b_n}{b_n}, 1)) - P_I^i), 1)) \quad (8)$$

$$P_B^i = \max(0, \min((\max(0, \min(\frac{P_B B - \sum_{n=1}^{i} P_B^n b_n}{b_n}, 1)) - P_I^i - P_P^i), 1)) \quad (9)$$

The video quality depends on the number of lost packets and on the content of the missing packet. The loss of a B-frame packet will not reduce the quality so significantly than an I-frame loss, due to frame dependencies of the MPEG video stream structure. The aim of the proposed scheme is to reduce the I- and P-frame data losses.

The number of lost I-, P- and B-frame packets can be estimated based on the link packet loss probability (π_i) and the previously introduced frame ratio ratios on the links, see equations (7)–(9). Considering that the interfaces are ordered according to the links' packet loss ratio ($\pi_1 \le \pi_2 \le \dots \le \pi_n$) the number of lost packet, which contains I-frame data, can be calculated as follows:

$$N_I = \sum_{i=1}^{n} \frac{P_I^i b_i}{S} \pi_i \qquad (10)$$

The packet size is signed with S, while b_i stands for the buffer size of the ith interface. Similarly to (10), the number of P- and B-frame packet losses can be also estimated.

III. PERFORMANCE EVALUATION

This section presents and discusses the performance of the proposed algorithm, and compares our content-aware interface selection solution to *continuous* and to *balanced* interface selection method.

The *continuous* method fills the interface buffers continuously from the video source or the video data storage. This model does not use heap up buffer, therefore the advantage of this method that no extra delay is introduced. We consider that first the most reliable link buffer is filled and if the link's capacity is fully utilized, the next reliable interface is selected and so on. The distribution of different frame type data on the links will be similar to the original video frame structure ($P_I : P_P : P_B = P_I^i : P_P^i : P_B^i$). Previous works does not apply heap up buffer, hence the actual frame data must be forwarded to the link buffers immediately.

The *balanced* method fills the interface buffers uniformly,

using the Round-Robin model. If of one of the interface buffer is full, the remained video data is distributed between the other interfaces, which are not full yet. This Round-Robin based interface selection method does not take the link characteristics into account. The distribution of the video frame type data in the interface buffers will be similar to the original video frame structure ($P_I : P_P : P_B = P_I^i : P_P^i : P_B^i$).

We have developed a simulation tool to analyze the behavior of the proposed content-aware interface selection scheme. Each simulation scenario was executed ten times and an average was calculated from the obtained results.

The *mother_and_daughter* QCIF sequence was used as the reference video. The video sequence was compressed with an MPEG-4 encoder – *ffmpeg* [16]. The video stream was coded at 25 fps and 300kbps. The frame structure is specified by the $N=10$ and $M=5$ parameters. N and M are the intraframe and interframe coding ratios, which define the sequence of I-, P- and B-frames. N specifies the I-frame interval whereas M determines the I- or P-frame interval. The distribution of the used video data bytes belonging to I-, P- and B-frames is P_I=51%, P_P=20% and P_B=29%.

In the analyzed test scenarios six links were applied with different characteristics. The link properties (bandwidth and loss ratio) used in the simulations are given in Table I.

TABLE I
LINK PROPERTIES

Link no.	Bandwidth [kbps]	Packet loss ratio
1	100	0.0
2	80	0.001
3	80	0.005
4	60	0.01
5	60	0.01
6	20	0.05

The first point of our performance examinations was to analyze the link utilization ratio. We have examined how the proposed interface selection method utilizes the available resources. In our method a heap up buffer is used to collect definite number of frames (causing definite delay). When more data is collected in the heap up buffer, the packet distributor can be more effective, because it can found frames with small sizes, which can still fit to the nearly full interface buffer. In the first test scenario we have compared the link utilization of the proposed method with different heap up buffer sizes. The smallest examined buffer size was able to store 20 frames, which is equal to 0.8 seconds delay, while the biggest heap up buffer had 160 frames (d_B=6.4s). The interface loads are presented in Fig.4.

The results show that setting the heap up buffer high, the usage of most reliable channels is also higher, while the load of less reliable ones is lower compared to other heap up buffer settings. For example, when the packets are distributed from a 160 frame sized heap up buffer, the link with number 6 is even not used, because the packet distributor could found free gaps in the interface buffers more efficiently and filled the reliable link buffers.
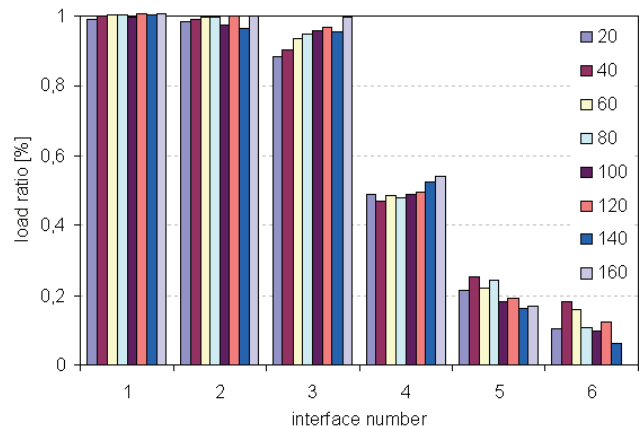

Fig. 4. Average interface buffer loads with different heap up buffer sizes

In the presented link utilization figure (Fig. 4.) some of the links are barely used, while on the links signed with number 1 to 3, closely the entire bandwidth is used. The utilization level of the links varied in time, because the heap up buffer level was also varying. The heap up buffer load is determinated by the number of frames (or delay) and not by the data size, therefore the data amount size in bytes can also change in time (see Fig. 5.).
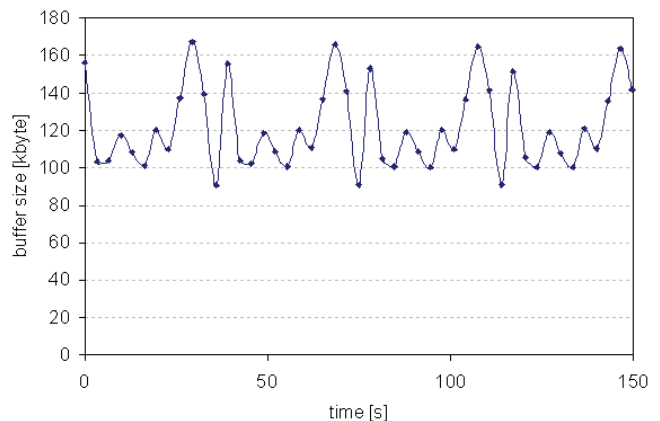

Fig. 5. Heap up buffer size variation

Low level of utilization can be seen on links with number 4–6 (Fig. 4.). This is due to average calculations, because the worst links are not used when the actual heap up buffer level is low. Similarly, if the current heap up buffer size is high, all the links are used.

After the link load measurements the MPEG frame type distribution in the interface buffers was analyzed. The performance of the proposed content-aware interface selection scheme depends of the video frame distribution on the links with different characteristics. Our aim is to deliver the I-frame data packets on the most reliable channels and transmit the P- and B-frames on the reliable links if the link capacity is sufficient.
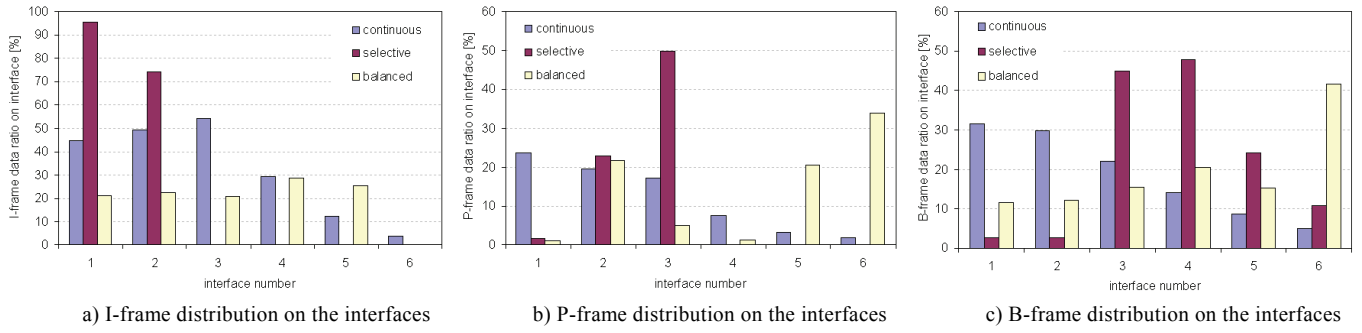
| a) I-frame distribution on the interfaces | b) P-frame distribution on the interfaces | c) B-frame distribution on the interfaces |

Fig.6. Different frame type distributions in the interface buffers

Fig. 6. illustrates the different frame type distributions in the interface buffers. In the analyzed scenario the heap up buffer size was set to 80 frames, which is equal to d_B=3.2s heap up delay. The interfaces are numbered based on the packet loss ratios and ordered, so the interface which uses the most reliable link has number 1, while the link with the highest drop probability has number 6. The video bitrate is 300kbps and the I-frame data ratio of the MPEG-4 video stream is P_I=51%, hence the I-frame data bandwidth requirement is about 150kbps. As the presented results show I-frame data can delivered on the first two interfaces (100kbps+80kbps) when the proposed content-based selective packet distributor algorithm is used (see Fig. 6a.)

Most of the P-frames were delivered on the 3th interface, while the B-frames were transferred on the less reliable channels. Due to this ordered packet delivery the number of lost packets belonging to I-frames is estimated to be significantly lower then the number of other frame packet losses.

The observed video quality highly depends on the number of lost packets and the packet content. In order to analyze the behavior of the proposed multi-path delivery method we have examined the different frame type data packet losses. The overall packet loss ratio was also measured, which is the average of the I-, P- and B-frame packet loss ratios. The results are presented in Fig.7. Similarly to the previous measurements the heap up buffer size was set to 80 frames.
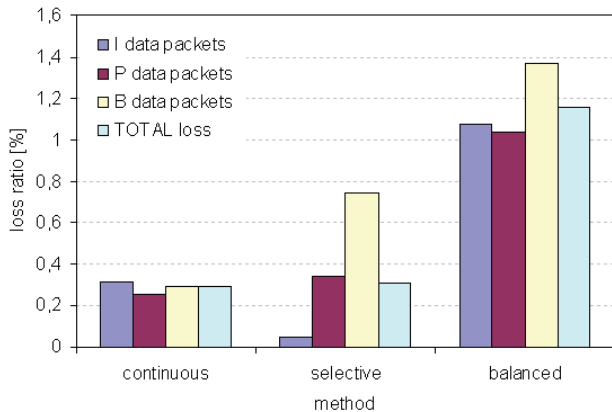

Fig. 7. Total and I-, P-, B-frame type data packet loss ratios

As we expected, the total number of packet losses was equal using the continuous packet distributor method and the selective one. In spite of the similarity the number of I-frame data packet losses was significantly lower when the proposed interface selection algorithm was used. In case of the continuous algorithm the loss probabilities of different type of frame data was near equal, however protecting the I-frame data the error propagation can be prevented, hence video quality improvement can be achieved. The worst method is the balanced one as Fig. 7. shows, because in this case the packet distributor equally fills the buffers belonging to reliable and unreliable channels.

The main goal of our proposal is to improve the quality of video streams. To see the effectiveness our scheme we measured the average peak signal-to-noise ratio (PSNR) of the transmitted video stream with different heap up buffer size setups. The PSNR values were measured against the original frames applying the content-aware interface selection scheme, the continuous and the balanced method. As it was expected, the measured PSNR value dropped dramatically when the packet loss increased.
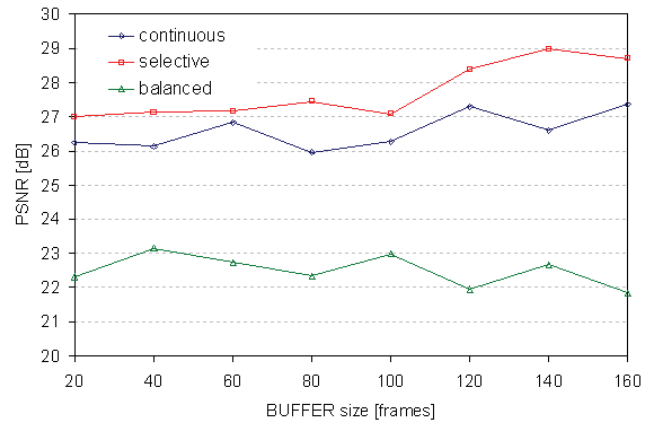

Fig. 8. Average PSNR measured with different heap up buffer size setups

The video quality difference between the selective and continuous scheme was not so significant because the total packet loss ratio in case of the two schemes was similar. The performance of the balanced method is very poor. When only the link loss probabilities were taken into consideration (continuous method) the video quality was acceptable.

When the packets are distributed based on its content, further quality improvement was achieved. The difference was about 1dB in our tests, but of course it can vary significantly according to the link properties and the coded video sequence.

IV. CONCLUSION

The increase of multimedia applications over the Internet has placed new requirements on current video streaming solutions. High bandwidth video provision in mobile environment can be also available if the mobile terminal is equipped with more interfaces. With the effective utilization of the various links the quality of the video stream can be significantly increased.

In this paper a content-aware interface selection scheme was presented, which provides efficient multi-path delivery of MPEG video streams. In our solution a heap up buffer is applied to increase the efficiency of the packet distributions, in order to transfer the high priority data on the reliable links. With this solution the overall packet loss is not decreased, however the I-frame data loss can be minimized and the error propagation in the video stream can be prevented. To evaluate our proposal simulations were performed and the obtained results proved the efficiency of our multi-path transmission scheme. Due to comprehensive modeling of video streams, the proposed low complexity solution provides an interesting and effective solution for video streaming in multi-path infrastructures.

ACKNOWLEDGMENT

REFERENCES

[1] L. Golubchik, J. Lui, T. Tung, A. Chow, and W. Lee, "Multi-path continuous media streaming: What are the benefits?," ACM J. Perform. Eval., vol. 49, no. 1-4, pp. 429–449, Sep. 2002.

[2] J. G. Apostolopoulos and M. D. Trott, "Path diversity for enhanced media streaming," IEEE Commun. Mag., vol. 42, no. 8, pp. 80–87, Aug. 2004.

[3] Pascal Frossard, Juan Carlos de Martin, Reha Civanlar, "Media Streaming with Network Diversity", Invited paper, Proceedings of the IEEE, vol. 96, no 1, pp. 39-53, January 2008.

[4] Y. J. Liang, E. Setton, and B. Girod, "Channel-adaptive video streaming using packet path diversity and rate-distortion optimized reference picture selection," in IEEE Fifth Workshop on Multimedia Signal Processing, 2002.

[5] S. Lin, S. Mao, Y. Wang, and S. Panwar, "A reference picture selection scheme for video transmission over ad-hoc networks using multiple paths," in IEEE ICME, 2001.

[6] R. Karrer, T. Gross, "Multipath streaming in best-effort networks", IEEE International Conference on Communications (ICC), Anchorage, Alaska, May 2003

[7] Jacob Chakareski, Pascal Frossard, "Distributed Collaboration for Enhanced Sender-Driven Video Streaming", IEEE Transactions on Multimedia, vol. 10, no 5, pp. 858-870, August 2008.

[8] Ahsan Habib, John Chuang, "Multihoming Media Streaming", IEEE Workshop on Multimedia Systems and Networking, WMSN 2005, April 2005.

[9] A. C. Begen, Y. Altunbasak, O. Ergun, and M. M. Ammar, "Multi-path selection for multiple description encoded video streaming", EURASIP Signal Processing: Image Comm., vol. 20, pp. 39-60, Jan. 2005.

[10] Dan Jurca, Pascal Frossard, "Video Packet Selection and Scheduling for Multipath Video Streaming", IEEE Transactions on Multimedia, vol. 9, no. 3, pp. 629-641, April 2007.

[11] R. Stewart (Ed.): Stream Control Transmission Protocol, IETF RFC 4960, September 2007.

[12] G. Ye, T.N. Saadawi and M. Lee: "IPCC-SCTP: an enhancement to. the standard SCTP to support multi-homing efficiently," IPCCC 2004, Phoenix, Arizona, April 15-17, 2004

[13] J. Iyengar, P. Amer, R. Stewart, "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths", IEEE/ACM Trans on Networking, 12/06.

[14] T. D. Neame, M. Zukerman, R. G. Addie, "Application of the M/Pareto Process to Modeling Broadband Traffic Streams", IEEE ICON '99, Brisbane, Australia, 1999.

[15] T. D. Neame, M. Zukerman, R. G. Addie, "Applying Multiplexing Characterization to VBR Video Traffic", ITC 1999 16, June 1999.

[16] FFmpeg Project, http://ffmpeg.mplayerhq.hu/