Session Management of a Variable Video Rate Streaming Session over Multi-Channel Networks

Rachel Behar
Faculty of Computer Science
Jerusalem College of Technology
Jerusalem, Israel
Email: rharris@jct.ac.il

Yael Samet
Intel
Jerusalem, Israel
Email: ygalinsk@jct.ac.il

Ronit Nossenson
Service Performance,
Akamai Technologies
Cambridge, MA, USA
Email: rnossens@akamai.com

***Abstract*—In this paper, we propose and evaluate two algorithms for session management of a variable bit rate video session over a multi-channel network. The session manager decides how many channels should be active in the next time interval on the basis of required video bit rate, session measurement reports and other considerations. The algorithms performance is evaluated against a fixed selection of the number of active channels. Simulation results reveal that it is possible to control the session costs in terms of the number of active channels while keeping the quality of the received video stream on the top Mean Opinion Score (MOS) level. System performance significantly improved in the feedback-based managed session, as compared to the simple-managed session and to the fixed selection of the number of active channels sessions.**

Keywords- multi-channel video transmission; session management; multimedia networks; video QoS.

I. INTRODUCTION

A video streaming application encodes, packetizes and transmits video frames in real-time. In other words, every streaming video frame needs to meet a play-out deadline. Currently, most networks support real-time services only in a best-effort manner. Therefore, video streaming services have to include special measures to be resilient to packet loss and late arrival. Over the last decade, streaming over multiple channels (also called multi-path, or networks) has been suggested to improve the video quality over the Internet [5][6][9]-[14], in peer-to-peer networks [15][19]-[21] and wireless ad-hoc networks [7][8]. Multi-channel video transmission is often coupled with adaptive/scalable layered-video encoding, e.g., H.264, Scalable Video Coding (SVC), to overcome channel rate variation and heterogeneous video client capabilities. Using multiple channels in layered-video transmission has also led to new challenges, such as video packet scheduling and new multi-channel encoding schemes [4][8][16]-[18].

In this study, we explore the contribution of the session management module in a multi-channel variable bit rate video session. Assuming that M channels can be activated in a particular video session, the session manager decides on the number of active channels $A \leq M$. The decision is based on the required video bit rate, and on the session measurement feedback reports regarding the channel conditions from the beginning of the session up to the last time interval. Additional considerations affecting this decision include the required video Quality of Service (QoS) parameters, information regarding the Quality of Experience (QoE) parameters, etc. The number of active channels corresponds to target performance indicators, such as target error rate in the next time interval. We focused on the minimal number of active channels that satisfy the performance requirements in order to reduce the overall system overhead. The session management algorithms described in this study have the following properties: (i) Simple decision function; (ii) Low computation afford; (iii) Small state and storage requirements; and (iv) Little channel feedback information (used only by the second algorithm).

Recently, an algorithm for session management of multi-channel constant rate video streaming session over wireless networks was suggested in [3]. We propose and evaluate two enhanced algorithms that support a variable bit rate video session. The evaluation of the management algorithms is based on simulation environment. The simulation results show that it is possible to control the session costs in terms of the number of active channels, while keeping the quality of the received video stream in the top mean Opinion Score (MOS) level. For example, comparing with static selection of three channels, the simple management algorithm achieved a 13.67% percent cost reduction with 2.59 active channels on average, and average Peak Signal-to-Noise Ratio (PSNR) of 37.32 comparing to average PSNR of 38.14 (reduction of only 2.15%). The feedback-based management algorithm achieved an 8.67% percent cost reduction with 2.74 active channels on average. Furthermore, the feedback-managed algorithm delivered 89.28% bytes on-time compared to only 84.66% bytes that were delivered on-time using a static selection of three channels. This leads to an average PSNR of 38.51, that is, a 0.97% percent improvement in the average PSNR compared to the static selection of three channels. Using a more sophisticated decoder could further improve the PSNR.

This paper is organized as follows: In the next section, the problem statement and rationale for session management are discussed. The simple management and feedback-based management algorithms for a variable bit rate multi-channel video streaming session are described in Section III. In Section IV, we present our simulation environment. In Section V, the simulation results are reported. Conclusions and further research directions are discussed in the last section.

 ISBN: 978-1-61208-435-0

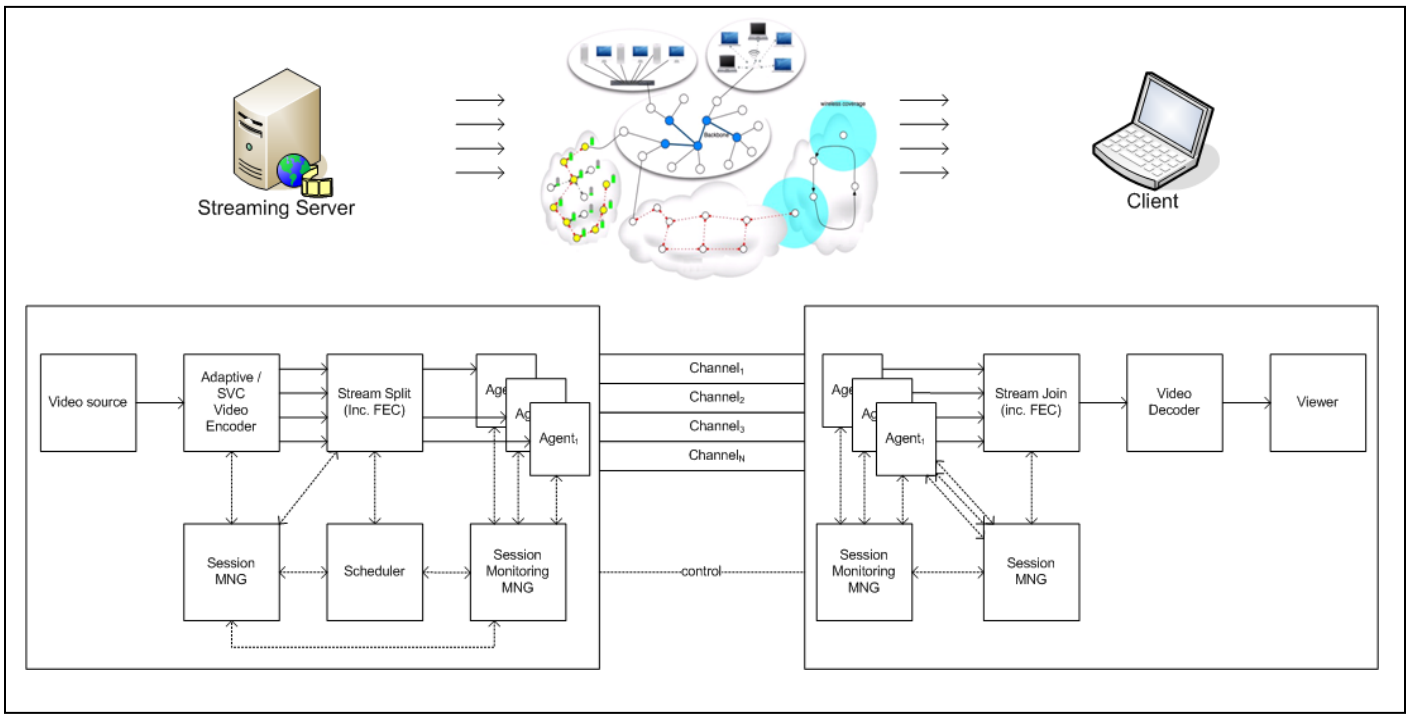

Figure 1. Architecture of the multi-channel video streaming system [3]

II. MULTI-CHANNEL VIDEO SYSTEM

In this section, we provide an overview of the multi-channel video transmission system under consideration [3]. As shown in Figure 1, the system consists of three parts: the video server, the multiple channels and the video client. The server and clients can communicate over up to *M* multiple channels (paths, networks). The video server consists of a video source, a video encoder, a module for stream splitting and channel protection, a session monitoring module, channel scheduler and a session manager module. The video client consists of a module for joining and decoding the channel protection, a session monitoring manager module, a session manager module, a video decoder and a viewer. These components are described briefly in the following paragraphs.

We assume that a space-time discrete video signal is used as input to the layered video encoder, which is characterized by its operational distortion-rate function. After source coding, the compressed layered video stream is prepared for transmission by the channel codec. This involves packetization and Forward Error Correction (FEC) combined with interleaving to reduce the effect of burst errors. After channel encoding, the video layers are scheduled to active channels and then the video packets are transmitted over the channels according to their layer to channel mapping.

In a general multi-channel network, different channels may have different parameter values. Furthermore, channel parameters may change due to the activation of other channels that share some resources, such as bottlenecked links [22]. We used the enhancement of the Gilbert-Elliott model [1][2] into a packet erasure multi-channel model [22] to characterize the multi-channel behavior in terms of video rate and error rate. According to this model, the multi-channel video rate for homogeneous channels is generated using the following formula:

$$\text{One channel:} \quad R_1 = R \cdot (1-\alpha_1)$$

$$\text{Two channels} \quad R_2 = 2R \cdot (1-\alpha_2)$$

$$\cdots$$

$$M \text{ channels} \quad R_M = MR \cdot (1-\alpha_M)$$

Where $\alpha_1 < \alpha_2 < \cdots < \alpha_M$. That is, the error rate increases with the number of active channels [22].

III. ALGORITHMS FOR SESSION MANAGEMENT OF MULTI-CHANNEL VARIABLE BIT RATE VIDEO STREAMING SESSION

The session manager's tasks are:

1) *Calculate: target video rate and additional parameters.*
2) *Decide: the number of active channels A ($1 \le A \le M$)*

In this paper we focus on the question of deciding the number of active channels.

In this section, we propose two session management algorithms in a multi-channel video system for transmitting video with variable video rate. The first algorithm is a simple algorithm and the second one is a feedback-based algorithm. These algorithms are compared with each other and with static sessions (unmanaged), in which the number of active channels is constant.

 ISBN: 978-1-61208-435-0

The simple session management procedure is as follows.

```
Void SimpleSessionMNG_Procedure()
Begin
1:  In the first time interval Do:
1.1:   Initiate session;
1.2:   Activate all M channels;
2:  In the second time interval Do:
2.1:   Get report from session monitor;
3:  For each time interval i>1 Do:
3.1:   Calculate target video rate R_V;
3.2:   Activate A+1 channels such that  R_A ≥ R_V;
3.3:   Update the other modules;
End
```

The simple-managed module initiates the session. In the first time interval, the algorithm activates all *M* channels. The algorithm gets a measurement report only once (after the first time interval), and learns the effective bandwidth of each channel from this report. In the following time intervals, the algorithm decides the number of active channels according to the target video rate of each time interval and the effective bandwidth of the channels that was reported at the beginning of the session. The channels are chosen sequentially. Either by index number (arbitrarily defined), or by descending order of bandwidth, that was estimated based on the report of the first interval.

The feedback-based session management procedure is as follows.

```
Void FeedbackSessionMNG_Procedure()
Begin
1:  In the first time interval Do:
1.1:   Initiate session;
1.2:   Activate all M channels
2:  For each time interval i>1 Do:
2.1:   Get report from session monitor;
2.2:   Calculate target video rate R_V;
2.3:   Decide the number of active channels A;
2.4:   Update the other modules;
End
```

The feedback-managed module also initiates the session and activates all *M* channels in the first time interval. Afterwards, in each time interval, the algorithm calculates the target video rate, decides the number of active channels and chooses the particular set of channels. Finally, it updates the other modules.

The next procedure describes the algorithm for deciding the required number of active channels according to session history and the target video rate in the next time interval (step 2.3 of the feedback-managed procedure).

```
Int FeedbackManagedActiveChannels (R_V, M,
                  timeInterval, channelSize[])
Begin
1:  Calculate last interval arrival percent
       based on monitor report;
2:  For each channel i in M channels Do:
2.1:  set availableChannelSize[i] =
       channelSize[i]*lastIntervalPercent[i];
3:  If Σ_M availableChannelSize < R_V
3.1:  Return M;
4:  Find min A such that
       Σ_A availableChannelSize ≥ R_V
5:  Return A+1;
End
```

In each time interval, the algorithm gets a measurement report of the active channels from the session monitoring manager and calculates the percent of the data that arrived on time out of the sent data. Then the current available bandwidth for real-time transmission of each channel is determined. Finally, the algorithm finds the minimum number of channels whose available bandwidth sum is sufficient for the target video rate in the next time interval.

The task of trying to find the minimum number of channels that satisfies the video rate (step 4 of the FeedbackManagedActiveChannels procedure) can be performed in several ways that differ by complexity and accuracy. Different methods can provide different results. We suggest two possible methods: a simple sequential method and a more complex optimal method. The first method is described in the next procedure.

```
Int FindSequentA_Procedure(R_V)
Begin
1:  For k=1 to M
1.1:  If Σ_{i=1}^{k} availableChannelSize ≥ R_V
1.1.1:     return k;
End
```

The FindSequentA procedure chooses the channels sequentially. In each iteration, the procedure tests the channels indexed 1 to k. If the sum of their available size is enough, the first k channels are chosen. The sequential method is very simple to compute, but does not always provide the best minimal set of channels. The second suggested method is described below:

```
Int FindOptimalA_Procedure()
Begin
1: For n=1 to M
1.1:   Find subset N of n channels such that:
       Σ_N availableChannelSize Is maximal;
1.2:   If Σ_N availableChannelSize ≥ R_V
1.2.1:          return n;
End
```

The FindOptimalA procedure uses exhaustive search to choose the set of *A* channels as finding such subset is NP-complete [29]. In each iteration, the procedure tests all $\binom{M}{n}$ subsets of *n* out of the *M* possible channels, $1 \le n \le M$, and finds the set with maximal bandwidth. If that set (with maximum available channels size) has enough bandwidth, that set is chosen. The method gives optimal results by examining sets of channels of varying sizes. It starts with sets of one channel, then tests sets of two channels until the set of *M* channels. When the procedure finds a sufficient set – that set is chosen. This order of testing the sets ensures the optimal choice with minimum number of channels and maximum bandwidth.

 ISBN: 978-1-61208-435-0

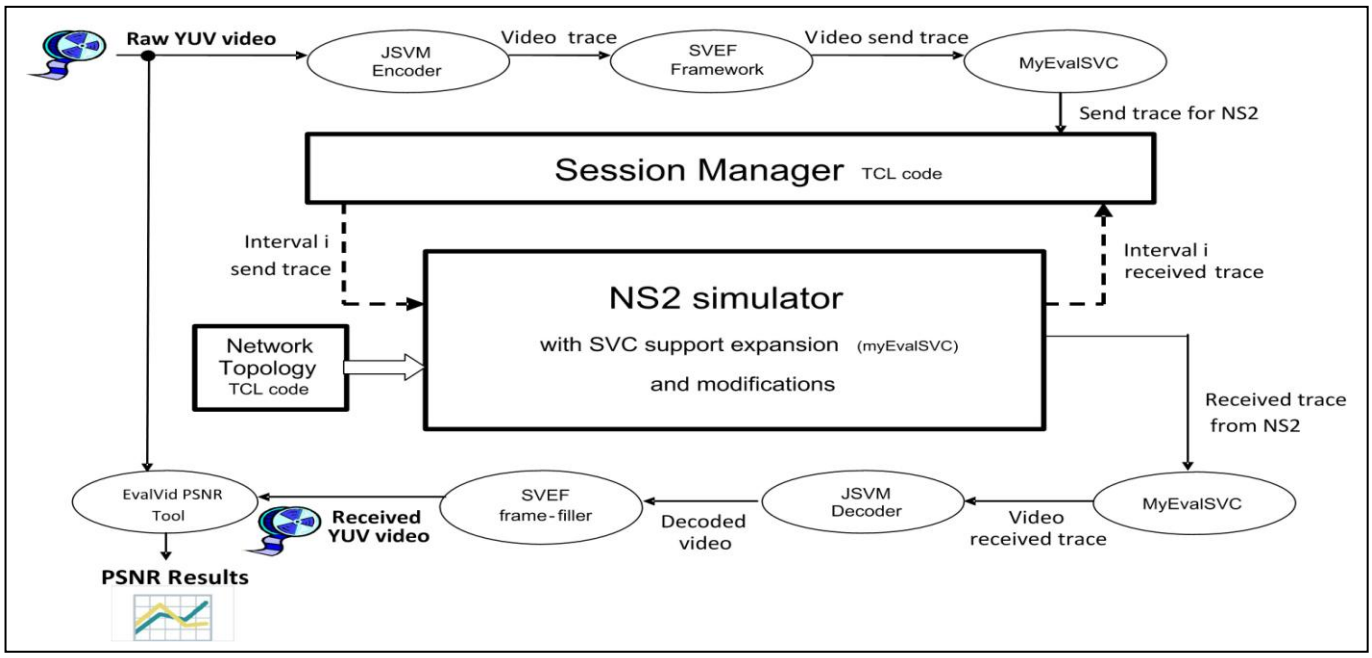

Figure 2. The Simulation Environment

IV. IMPLEMENTATION WITH NETWORK SIMULATOR

In this section, we describe the simulation environment used for evaluating the suggested session management algorithms performance. The environment is Network Simulation – 2 (NS-2) based. NS-2 is an open-source network simulator widely used in academic research [23]. The simulator is fed with scripts and traces to be sent over the simulated network. It simulates the network behavior and generates traces for the receiving end. We implemented the session-manager algorithm in TCL and ran it on NS-2. In addition, we used the following open-source tools:

- The JSVM Reference Software [24] - the reference software for the Scalable Video Coding (SVC) project of the Joint Video Team (JVT).
- The SVEF Framework [25] – a Scalable Video coding streaming Evaluation Framework, devised to evaluate the performance of H.264 SVC video streaming.
- The MyEvalSVC [26] - an integrated simulation framework for evaluation of SVC transmission based on the SVEF and extended to connect to the NS2 simulator.
- The EvalVid [27] - a tool-set developed for evaluation of the quality of a video transmitted over a real or simulated network.

In Figure 2, the simulation environment is presented. As can be seen from this figure, a raw YUV video is encoded using the JSVM-Encoder and a video trace file is created by JSVM's Bit Stream Extractor. The trace file is processed in the SVEF's f-nstamp tool and a send-trace file is generated. The send-trace is converted to the format of a NS2-send-trace file using MyEvalSVC tools. The full NS2-send-trace is delivered to the session manager module. In each time interval (one second of video), the manager splits the interval's part of the NS2-send-trace into *A* different NS2-send-traces, according to the algorithm decision of *A* active channels. Then, NS-2 simulates sending the video interval over *A* channels out of the *M* channels we defined in the network topology. When the interval is completely received on the receiver end, the session manager module gets the received-trace, and based on the performance that is derived from the send-trace and received-trace, it decides the number of active channels for the next interval. The manager repeats this routine until the entire video is transmitted.

When the simulation is completed, the full NS2-received-trace of the video is converted into a video trace using MyEvalSVC tools. The JSVM-Decoder decodes the received video trace back to a raw YUV video file. Since some frames might be lost during the transmission, the video is reconstructed using the SVEF frame-filler tool, which fills missing frames by duplicating other frames, so that the received video has the same length as the sent video. Finally, the quality of the received video is evaluated according to its frames PSNR value, which is calculated by comparing it to the original video with EvalVid's PSNR tool.

V. SIMULATION RESULTS

In this section, we describe the benchmark, the simulated network topology, and the performance of the session management algorithms.

To validate the session manager's performance, we decided to use the PSNR matric that measures video quality. The PSNR value is calculated by comparing two raw YUV formatted video sequences. Therefore, we searched for long original YUV sequences. Most of the video traces include short video only, and were not suitable to study the management

 ISBN: 978-1-61208-435-0

algorithms benefit and limitations. The most suitable sequence we found was the open source animation video Big Buck Bunny [28], a long high-quality YUV sequence. We simulated transmitting 5 minutes of the video. The frame rate of the video was 24 fps and we defined time interval length to be one second of video. The total number of transmitted intervals was 300 and the number of frames was 7200.

The transmission of the video was tested over two different topologies. The first topology was very simple. The source and the destination were connected with five independent channels. In this case, there was no significant difference between the different management algorithms. In reality, in a multi-channel network, independent channels are very uncommon. Hence, we tested the second network topology as presented in Figure 3 that better reflects realistic multi-channel conditions. In this network topology, the source is connected to the destination with five channels as well. But, the first two channels are occasionally disturbed by other entities that use their resources (source 2 transmits to destination 2 and source 3 transmits to destination 3) and channels three and four correspond to two edge dependent paths. The fifth channel is independent. The propagation delay of each link is 1ms, and the bandwidth is 4Mbps.

We simulated both simple and feedback managed algorithms, each one with the two methods of channels selection we described. The results of both versions of the simple manager were very similar, hence we will present only the results of one method for the simple manager. We assume that sorting the channels (the second method) does not improve the results compared to sequentially choosing the channels (the first method), because the simple manager gets report only after the first interval, and that report does not sufficiently reflect the channels' capacity that changes over time.

Figure 4, 5, and 6 present the simulation results of the feedback-managed and simple-managed algorithms. Throughout this section, we refer to *information loss* in case of either actual information loss or in case of late arrival (according to the video play-time). Figure 4 presents the results of the byte-loss percent (a), and the video frame-loss percent (b) for each time interval for the optimal and sequential feedback-managed algorithms (green and red lines) and simple-managed algorithm (blue line). We can see in both graphs that the optimal feedback-managed algorithm lost the least amount of data, the sequential feedback-managed algorithm lost more data than the optimal and the simple-managed algorithm lost the most amount of data.

We refer to the loss of data in two aspects: byte-loss and video frame-loss. The difference between the two aspects is due to the simple open source decoder that decodes the received video. The decoder is not sophisticated enough to handle decoding of partial frames, therefore even if only a few bytes are missing, a whole frame is deleted and possibly other frames depending on this frame. For example, in time interval number 229 (marked with a dashed line number 1 in Figure 4), the difference is noticeable. The red graph that represents the loss of the sequential feedback-managed algorithm in very low (0.9%) in graph (a) (byte-loss), but is significantly higher (9 out of 24 video frames) in graph (b) (frame-loss). An additional example that emphasizes the difference is provided in time interval 35 (marked with dashed line number 2). In graph (b),

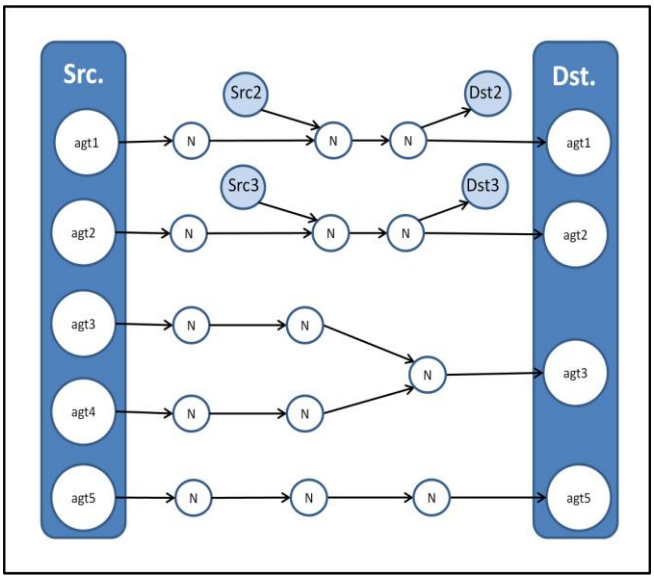

Figure 3. Simulated network topology

the red and blue graphs (sequential feedback-managed and simple-managed respectively) are very close, meaning they lost almost the same number of video frames (11 lost video frames using the feedback-managed algorithm vs. 12 lost video frames using the simple-managed algorithm). However in graph (a) at the same point, the difference between the two graphs is distinguishable (17% byte loss vs. 33% byte loss).

Figure 5 provides an example of the performance results with respect to the PSNR. The results of frame-loss (a) and PSNR value (b) for the two feedback-managed and simple-managed algorithms in time intervals 75-95 are plotted in this figure. The frame loss was calculated per interval, while the PSNR was calculated per frame, so that each point in graph (a) that represents one interval is equivalent to a sequence of 24 frames in graph (b). Clearly, high video frame loss will cause low PSNR. The PSNR value is calculated by comparing the original YUV sequence to the YUV sequence that was decoded from the received SVC video. The PSNR value is affected by two factors. First, the encoding and compressing of the original video by the video encoder before it was sent over the simulated network (from YUV to SVC). The second factor is the data loss caused by the video transmission over unreliable channels. All algorithms were affected identically by the first factor. The difference in the PSNR between them was caused only by the second factor, hence, the difference between the graphs points out the advantage of both of the feedback-managed algorithms over the simple-managed algorithm, and the advantage of the optimal method over the sequential method. In time intervals 75-76, 92-95, there was no loss of data in all algorithms, as can be seen in graph (a), and the PSNR value (shown in graph (b)) is high because it was affected only by the encoding process. In time intervals 77-80, the same number of frames were lost in the sequential feedback and the simple algorithms and the PSNR value is also identical, but the optimal feedback managed lost a lot less frames in intervals 79-80 and its PSNR value is much higher. In the rest

 ISBN: 978-1-61208-435-0

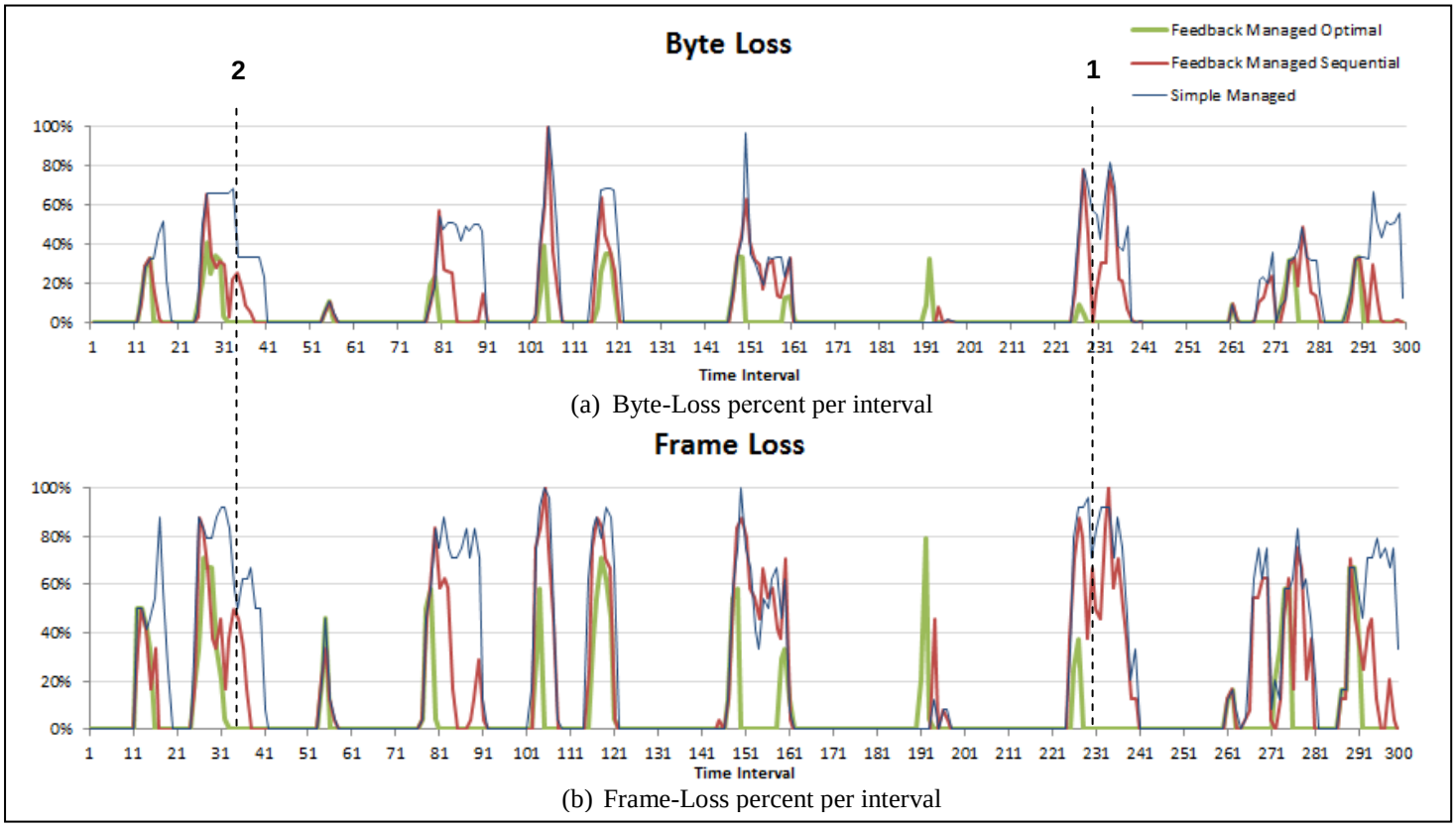

Figure 4. (a) Byte-Loss; (b) Video Frame-Loss for simple-managed (blue line), optimal (green line) and sequential (red line) feedback-managed algorithms

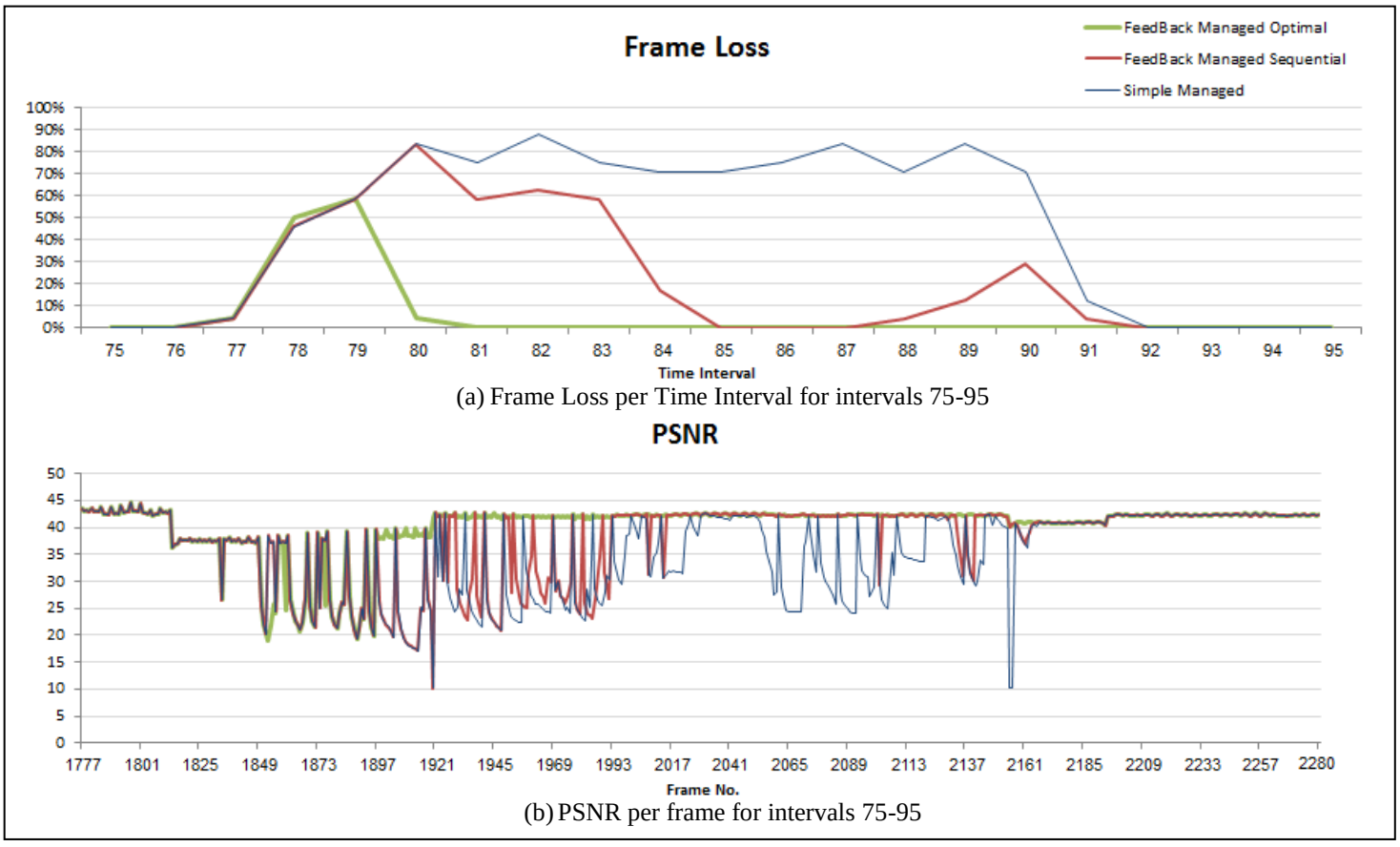

Figure 5. (a) Video Frame Loss; (b) PSNR for intervals 75-95 for simple-managed (blue), optimal (green) and sequential (red) feedback-managed algorithms

 ISBN: 978-1-61208-435-0

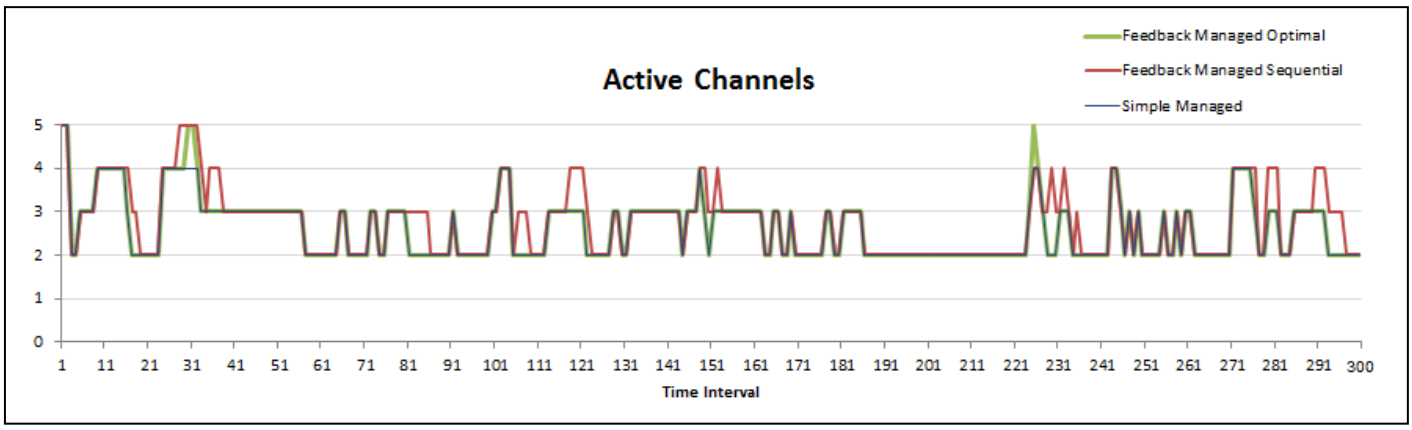

Figure 6. Number of active channels per time interval for simple-managed (blue), optimal (green) and sequential (red) feedback-managed algorithms

of the time intervals the optimal feedback-managed (green line) did not lose any frames and its PSNR value is the highest. In time intervals 81-84, graph (a) shows that the sequential feedback manager (red line) lost less video frames than the simple-manager (blue line), and graph (b) also shows that the PSNR of the sequential feedback managed is slightly higher than the simple-managed. In time intervals 85-91, the recovery of the sequential feedback-managed is clearly seen, in graph (a) its frame loss is very low and in (b), its PSNR is high, in contrast to the simple-managed algorithm whose frame loss remains high, and PSNR remains low.

Figure 6 plots the number of active channels during the simulation period. Due to the changes in the channels conditions that are reported only to the feedback-managed algorithms, there are intervals where the feedback-managed algorithms activates more channels than the simple-managed algorithm, as seen in the graph. The graph also shows that frequently the optimal feedback manager (green line) activates less channels than the sequential feedback manager (red line), because it chooses the channels with the biggest capacity, and therefore is satisfied with fewer channels.

TABLE I. PERFORMANCE EVALUATION

Method	Byte Loss	Active Channels Avg.	PSNR		PSNR violation
			Mean	Std.	
Feedback-Managed-OptimalA	5.13%	2.6	40.01	4.77	12.15%
Feedback-Managed-Sequential	10.72%	2.74	38.51	6.74	20.86%
Simple-Managed-Bandwidth-order	18.09%	2.59	37.35	7.82	27.34%
Simple-Managed-Sequential	17.68%	2.59	37.32	7.83	27.52%
Static 2-channels	46.72%	2	31.84	11.82	48.5%
Static 3-channels	15.44%	3	38.14	7.06	22.95%
Static 4-channels	11.89%	4	38.9	6.36	18.37%
Static 5-channels	4.92%	5	40.04	4.74	11.22%

Table I summarizes the performance evaluation of the two suggested session management algorithms and static sessions with 2,3,4,5 channels. A Static i-channels method (lines 3-6 in Table I) is a simple un-managed method in which the first i'th channels are selected to transmit the video regardless of the target video rate and the channels temporal conditions. For each method, we present in the table the results of: (i) Byte loss percent – the percent of the data that was lost in the transmission or arrived too late (considering a 1000ms play-out buffer). (ii) Active channels average – the average number of channels that were activated per time interval. (iii) PSNR – the average value of PSNR per frame and the standard deviation of PSNR per frame. The PSNR average alone does not fully reflect the quality of the received video, because if the average is high but the standard deviation is also high, the viewing experience is impaired due to the significant changes in the PSNR value between frames. (iv) PSNR violation – the percent of frames whose PSNR value is lower than 37, which represents a minimum threshold for high video quality according to MOS mapping.

As expected, the static 5-channels method outperforms the other method with high average PSNR and low PSNR std. However, since it activates five channels, it has the highest cost (assuming that every channel has its own operation costs). The static 2-channels method has the lowest performance and the lowest cost. Almost half of the information did not reach the destination on time, and it violates the top MOS PSNR level on 48.5% percent of the video frames. Between these performance edges, we have the static 3-channels, 4-channels and the two managed methods. It can be seen that the feedback-manage method successfully balances between cost, on-time information delivery and PSNR results. Note that although the static 4-channels method outperforms the feedback-managed method with slightly higher PSNR and slightly lower PSNR violation, the feedback-managed, in fact, delivered more information to the destination on-time and the gap is a result of the simple decoder in use in the simulation only.

 ISBN: 978-1-61208-435-0

VI. CONCLUSION

In this study we suggested two simple algorithms for session management in multi-channel variable bit rate video transmission. The algorithms have the following properties: they are very simple to compute, they require low computation afford, low storage requirements, and small feedback messages that provide the statistics for each active channel in the previous time interval (used only by the second algorithm).

The evaluation of the management algorithms is based on simulation environment. The simulation results show that it is possible to control the session costs in terms of the number of active channels while keeping the quality of the received video stream in the top MOS level. For example, compared to the static selection of three channels, the simple management algorithm achieved a 13.67% percent cost reduction with 2.59 active channels on average, and average PSNR of 37.32 compared to the average PSNR of 38.14 (reduction of only 2.15%). The feedback-based management algorithm achieved an 8.67% percent cost reduction with 2.74 active channels on average. Furthermore, the feedback-managed algorithm delivered 89.28% bytes on-time compared to only 84.66% bytes that were delivered on-time using a static selection of three channels. This leads to an average PSNR of 38.51, that is, a 0.97% percent improvement in the average PSNR compared with a static selection of three channels. Using a more sophisticated decoder could improve the PSNR even more.

ACKNOWLEDGMENT

We thank Net-HD video team and in particular Noam Amram for useful discussions. We thank Mordechai Behar for technically supporting this research.

REFERENCES

[1] E. O. Elliott, “Estimates of Error Rates for Codes on Burst-Noise Channels”, Bell System Technical Journal 42, 1963, pp. 1977–1997.

[2] E. N. Gilbert, “Capacity of a Burst-Noise Channel”, Bell System Technical Journal 39, 1960, pp. 1253–1265.

[3] R.Nossenson and N. Amram, "Session Management in a Multi-Channel Video Streaming System for Wireless Networks", The 18th IEEE International Conference on Networks (ICON 2012), Singapore, Dec. 2012, pp. 333-338.

[4] R.Nossenson and N. Amram, "Packet Scheduling in Multi-Channel Layered-Video Streaming over Wireless Sensor Networks", To appear, The 1st International Workshop on Wireless Multimedia Sensor Networks (WMSN'12) (part of WiMob 2012), Barcelona, Spain, Oct. 2012, pp. 683-688.

[5] R. Nossenson and O. Markowitz, “Using Coordinated Agents to Improve Live Media Content Transmission”, In proceedings of the Sixth International Conference on Systems and Networks Communications (ICSNC 2011) , Barcelona, Spain Oct. 2011, pp. 167-170.

[6] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee. "On multiple description streaming with content delivery networks", In IEEE INFOCOM, 2002, pp. 1736-1745.

[7] E. Setton, X. Zhu and B. Girod, “Minimizing distortion for multipath video streaming over ad hoc networks”, IEEE Int. Conf. Image Processing (ICIP-04), Singapore, vol 3, Oct. 2004, pp.1751–1754.

[8] S. Mao, S. Lin, S. S. Panwar, Y. Wang, and E. Celebi, "Video Transport Over Ad Hoc Networks: Multistream Coding With Multipath Transport", IEEE journal on selected areas in communications, Vol. 21, No. 10, Dec. 2003, pp. 1721-1737

[9] B. Wang, W. Wei, and D. Towsley, "Multipath Live Streaming via TCP: Scheme, Performance and Benefits", ACM Transactions on Multimedia, 2009, pp. 11.1-11.12.

[10] D. Jurca and P. Frossard, “Distributed media rate allocation in multipath networks”, Signal Process.: Image Commun., Vol.23, 2008, pp.754–768.

[11] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee. "On multiple description streaming with content delivery networks", In IEEE INFOCOM, 2002, pp. 1736-1745.

[12] L. Golubchik et al, “Multi-path continuous media streaming: What are the benefits?”, Performance Evaluation, 2002, pp. 429-449.

[13] Y. J. Liang, E. G. Steinbach, and B. Girod, “Real-time voice communication over the Internet using packet path diversity”, In ACM Multimedia, Ottawa, Canada, Sep. 2001, pp. 431-440.

[14] T. P. Nguyen, and Z. Avideh, “Mutiple sender distributed video streaming”, IEEE Transaction on Multimedia, 6(2), April 2004, pp. 315-326.

[15] M. Wang, L. Xu, and B. Ramamurthy, "Linear Programming Models For Multi-Channel P2P Streaming Systems", Mini-Conference at IEEE INFOCOM 2010, pp. 1-5.

[16] E. S. Ryu, H. Kim, S. Park, and C. Yoo, "Priority-Based Selective H.264 SVC Video Streaming Over Erroneous Multiple Networks", 2011 IEEE International Conference on Consumer Electronics (ICCE), 2011, pp. 337-338.

[17] L. Zhou, X. Wang, W. Tu, G. M. Muntean, and B. Geller, "Distributed Scheduling Scheme for Video Streaming over Multi-Channel Multi-Radio Multi-Hop Wireless Networks", IEEE journal on selected areas in communications, Vol. 28, No. 3, April 2010, pp. 409-419.

[18] L. Zhou, B. Geller, B. Zheng, S. Tang, J. Cui, and D. Zhang, "Distributed Scheduling for Video Streaming over Multi-Channel Multi-Radio Multi-Hop Wireless Networks", in proceedings of IEEE GLOBECOM 2009, pp. 409-419.

[19] V. Agarwal, and R. Rejaie, “Adaptive Multi-Source Streaming in Heterogeneous Peer-to-Peer Networks,” Proc. Multimedia Computing and Networking (MMCN ’05), Jan. 2005, pp. 13-25.

[20] Y. Guo, C. Liang, and Y. Liu, “AQCS: Adaptive Queue-based Chunk Scheduling for P2P Live Streaming,” in Proceedings of IFIP Networking, 2008, pp. 433-444.

[21] M. Zhang, Y. Xiong, Q. Zhang, and S. Yang, “Optimizing the throughput of data-driven peer-to-peer streaming”, IEEE Transactions on Parallel and Distributed Systems, vol.20, no.1, 2009, pp. 97-110.

[22] R. Nossenson, and N. Nossenson, “Packet Erasure Model for Multi-Channel Video Streaming”, to apear, IEEE International Symposium on Network Computing and Applications (NCA), 2015.

[23] NS2 Simulator http://www.isi.edu/nsnam/ns/ , retrieved Sep. 2015.

[24] JSVM Software, http://evalsvc.googlecode.com/files/SoftwareManual.doc, retrieved Sep. 2015.

[25] SVEF Fraemwork, http://svef.netgroup.uniroma2.it/ retrieved Sep. 2015.

[26] MyEvalSVC Toolset, https://code.google.com/p/evalsvc/ retrieved Sep. 2015.

[27] EvalVid Toolset, http://www.tkn.tu-berlin.de/menue/research/evalvid/ retrieved Sep. 2015.

[28] BigBuckBunny YUV Sequence, http://www.bigbuckbunny.org/ retrieved Sep. 2015.

[29] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, “Introduction to Algorithms” (3rd ed.). McGraw-Hill Higher Education, 2009.

 ISBN: 978-1-61208-435-0