Online multipath convolutional coding for real-time transmission

Tuan Tran Thai, Emmanuel Lochin, Jérôme Lacan
University of Toulouse; TéSA; ISAE/DMIA; Toulouse, France

***Abstract*—Most of multipath multimedia streaming proposals use Forward Error Correction (FEC) approach to protect from packet losses. However, FEC does not sustain well burst of losses even when packets from a given FEC block are spread over multiple paths. In this article, we propose an online multipath convolutional coding for real-time multipath streaming based on an on-the-fly coding scheme called Tetrys. We evaluate the benefits brought out by this coding scheme inside an existing FEC multipath load splitting proposal known as Encoded Multipath Streaming (EMS). We demonstrate that Tetrys consistently outperforms FEC in both uniform and burst losses with EMS scheme. We also propose a modification of the standard EMS algorithm that greatly improves the performance in terms of packet recovery. Finally, we analyze different spreading policies of the Tetrys redundancy traffic between available paths and observe that the longer propagation delay path should be preferably used to carry repair packets.**

I. Introduction

Multipath streaming has gained much attention recently thanks to overlay networks and multiple access technologies (e.g., Wi-Fi, Cellular) available by default in handheld devices. The benefits of multipath overlay routing and multipath streaming are presented in [1], [2] (e.g., reduction in correlation between consecutive packet losses, throughput gain, ability to react to congestion variation in different parts of the network). Another interesting property of multipath has been illustrated in [3]. Fashandi et al. [3] showed that the loss rate after packet recovery decays exponentially with the number of paths. However, the challenging task in multipath streaming is to split the data flow among available paths to achieve better perceived video quality. As a potential solution, in [4], Jurca et al. proposed a load splitting scheme based on an end-to-end (E2E) distortion model for single layer video streaming. Later in [5], they proposed a similar E2E distortion model for scalable video streaming as an objective function and used optimization algorithms to minimize the distortion. One of the most achieved algorithms is Encoded Multipath Streaming (EMS) framework proposed by Chow et al. [6]. In their proposal, the receiver observes the loss rate on each path, calculates the overall loss rate after packet recovery and sends the load splitting vector to the sender. However, all these proposals ([3]-[7]) use Forward Error Correction (FEC) to protect the video from losses. The main problem is this block code scheme requires to dynamically adapt its initial parameters and as a result, complex probing and network feedback analysis. Recently, a novel erasure coding approach that prevents such complex configuration has been proposed [8], [9], [10].

In this paper, we propose to use an on-the-fly erasure coding scheme called Tetrys [10] to real-time multipath streaming and in particular, inside the EMS framework [6]. The rationale of using this framework is because EMS obtains better performance in terms of computation compared to [4], [5] [11]. We show that enabling Tetrys instead of FEC inside EMS greatly improves the overall performance in terms of packet delivery ratio in both uniform and burst losses. We also study the decoupling between load allocation and redundancy traffic with Tetrys and propose several other measurements with different propagation delay not tackled in [6]. The results show that sending Tetrys repair packets on paths with longer propagation delay increases the packet delivery ratio before the E2E delay constraint in real-time transmission limited to hundreds milliseconds. Furthermore, we improve the EMS scheme to better follow the network dynamics and to reduce the loss rate after packet recovery.

The rest of this article is organized as follows. Section II introduces briefly the EMS scheme. Section III presents the basic principle of Tetrys and the decoupling between load allocation and Tetrys redundancy traffic. Section IV shows the results obtained from Tetrys compared to FEC with different settings and the benefits of decoupling between load allocation and Tetrys redundancy traffic. Section V presents the modified EMS algorithm and results. We conclude and provide future work in section VI.

II. EMS principle

Fig. 1 shows an overview of the EMS scheme. The FEC encoder in EMS sender takes the live stream and encodes with FEC parameters specified by k source packets and n-k repair packets. The encoded stream is then splitted among available paths with different characteristics (e.g., propagation delay, loss rate, available bandwidth) thanks to the packet scheduler. The EMS receiver stores all received packets and checks whether it can decode all lost packets in a FEC block specified by FEC(k,n). In the context of live streaming, any packets arrived or recovered after the deadline are discarded.

The EMS scheme is detailed in [6]. Thus, we introduce the most important part of EMS, the Online Load Splitting (OLS). At bootstrap, EMS sender equally splits the load between available paths so that the receiver measures the loss rate in each path. At each period defined by OLS Adapt Window (in second), the EMS receiver executes the OLS algorithm as depicted in the pseudo code 1. The information loss rate indicates the percentage of data that can not be recovered.

978-1-4673-0301-9/12/$31.00 ©2012 IEEE

After performing the OLS, EMS receiver sends a feedback containing the load splitting vector and the FEC parameters. The packet scheduler of EMS sender follows the load vector upon reception of the feedback.

Algorithm 1 Online Load Splitting (OLS)

```
 1: Compute the asymptotic optimal solution and split the load accordingly
 2: Sort the paths in the increasing order of loss rate
 3: repeat
 4:     Pick the first path in the list
 5:     repeat
 6:         Increase the load on the chosen path by pre-defined $\Delta_L$ (3% by default)
 7:         Decrease the load on each of remaining paths by a fraction of $\delta$, proportional to their respective loss rates
 8:     until measured information loss rate increases
 9:     Remove the chosen path from the list
10:     Revert to the previous load splitting
11: until the path list is empty
12: goto Step 2
```

It is noted that the load splitting vector only decides the amount of traffic that each path should carry. This inspires our study of decoupling between the load splitting vector and redundancy traffic with Tetrys (will be described in Section III-B).

III. Tetrys multipath

We introduce in this section an on-the-fly erasure coding scheme called Tetrys coupled with EMS scheme for real-time multipath streaming. Then, we present the rationale of decoupling between load allocation and Tetrys redundancy traffic.

A. Basic principle of Tetrys

Tetrys uses an elastic encoding window buffer B_{EW} which includes all the source packets sent without acknowledgment. For every k source packets, Tetrys sender sends a repair packet $R_{(i..j)}$ which is built as a linear combination of all packets currently in B_{EW} from packet indexed *i* to *j*. The receiver is expected to periodically acknowledge the received or decoded packets. Upon reception of acknowledgment, the sender removes the acknowledged packets out of its B_{EW}. Generally, the receiver can decode lost packets as soon as the number of repair packets received is equal to the number of lost packets. By this principle, Tetrys is tolerant to burst losses in neither source, repair nor acknowledgment packets as long as the redundancy ratio exceeds the packet loss rate (PLR). Furthermore, the lost packets are recovered within a delay that does not depend on the Round Trip Time (RTT). This property is very important for real-time applications.

Let us show in Fig. 2 a simple Tetrys data exchange with k = 2 which implies that a repair packet is sent for every two sent source packets (or redundancy ratio of 33.3%). The packet P_2 is lost during the data exchange. However, the reception of repair packet $R_{(1,2)}$ allows to rebuild P_2. The acknowledgment for packets P_1 and P_2 from the receiver is lost. This loss does not interrupt the transmission, the sender simply continues to compute the repair packets from P_1. Later, the lost packets P_3, P_4 are rebuilt thanks to $R_{(1..6)}$ and $R_{(1..8)}$. The reception of second acknowledgment packet allows the sender to remove the acknowledged packets and build the repair packets from P_9. The reader is referred to [10] for further details.

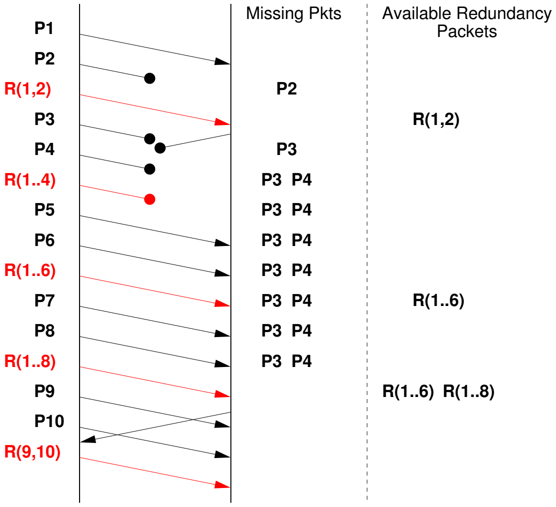

Fig. 2. A simple data exchange with Tetrys (k=2) [10]

B. Decoupling load allocation and redundancy traffic with Tetrys

In [12], Kurant showed that the propagation time differences between paths reach several tens of milliseconds by measurements. In case of FEC, the last packet (source or repair) in a FEC block arrived at the receiver must be sooner than the end-to-end (E2E) delay requirements normally specified by the path with longest propagation delay. If the arrival date of last FEC packet exceeds the deadline due to long block size or queuing delay in the network, the sender should reduce the block size. The size reduction makes FEC less tolerant to burst losses (see later in Fig. 6). Thus, we believe that FEC repair packets can be sent to any available paths without changing the result with well dimensioning block size. On the other hand, the arrival time of Tetrys repair packets is rather important since they are used to recover all previous lost packets if possible. If a repair packet built from sent source packets without acknowledgment is transmitted to the path with short propagation delay, it is likely that the repair packet arrives sooner than the source packets sent on longer paths. This means that the arrival of repair packet can not be used to recovered the previous lost packets at its arrival even though the source packets sent on longer paths arrive successfully. This reduces the effectiveness of Tetrys repair packets in real-time streaming. Based on this observation and the independence between load splitting vector and packet

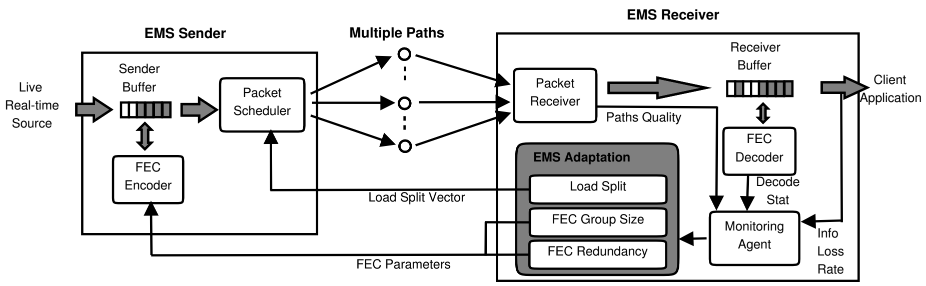

Fig. 1. EMS overview [6]

scheduler (see section II), we propose to decouple the load allocation on each path specified by the load splitting scheme and the way Tetrys repair packets are sent. This implies that Tetrys repair packets are preferably sent to the path with longer propagation delay while keeping the same load allocation.

Table I show different strategies to send Tetrys repair packets. For instance, with “Tetrys long” strategy, the Tetrys repair packets are first sent to the path with longest propagation delay. If the load on longest path is fulfilled, the Tetrys repair packets are sent to the path with second longest propagation delay and so on. While Tetrys repair packets are sent to the available path according to the packet scheduler in ”Tetrys” strategy.

TABLE I
DIFFERENT STRATEGIES OF SENDING TETRYS REPAIR PACKETS

Tetrys long	preferably sent to path with longer delay
Tetrys short	preferably sent to path with shorter delay
Tetrys	sent to available path

IV. SIMULATIONS AND RESULTS

We use *ns-2* [13] to evaluate Tetrys and FEC using EMS scheme. The number of paths is specified in each simulation. These paths can be built thanks to multiple physical interfaces or overlay network. The path establishment is out of scope of this article. We assume that the available bandwidth exceeds the application rate. The one-way E2E delay constraint is set to 150ms based on ITU-T/G.114 [14] which is recommended for highly interactive applications. One of the main characteristics of Tetrys is to be fully reliable whatever the burst size [10]. Indeed, all lost packets are recovered if the redundancy ratio exceeds the PLR. However, we consider the packets as lost at the application level if their arrival or recovery date exceeds the deadline. The information loss rate indicates the percentage of lost data that can not be recovered or be recovered after the deadline of 150ms. To simulate the burst losses, we use a Gilbert-Elliot model in [10] which is specified by an average PLR and an average length of consecutive lost packets (or shortly mean burst size). In each simulation, the streaming server sends a Constant Bit Rate (CBR) traffic at 1900 kb/s with packet size of 210 bytes. The frequency of Tetrys acknowledgment packet is set to 10ms. The feedback frequency does not change the result since it only affects the buffer sizes.

A. Comparison between FEC and Tetrys with the same EMS scheme

In this simulation, there are two paths between a sender and a receiver. The propagation delay on each path is set to 50ms. The streaming lasts 4 hours and the OLS Adapt Window is set to 60s. The redundancy ratio is set to 10% which is equivalent to FEC(45,50). The PLR on path 1 is set to 3% and the PLR on path 2 varies from 0% to 5%. Fig. 3 shows that Tetrys consistently outperforms FEC(45,50) in both uniform and mean burst size of 2 and 3 packets. More specifically, Tetrys reduces up to more than 1% information loss rate in case of mean burst size of 3 packets. With the video coding standard H.264/AVC, the Peak Signal to Noise Ratio (PSNR) with Tetrys can gain up to several dBs [15]. It is noted that the result with FEC(45,50) in Fig. 3 is similar to Fig. 14 in [6]. In fact, when the PLR on path 2 is less than 3%, the EMS scheme tends to assign more load on path 2, thus the information loss rate proportionally increases with the PLR on path 2. When the PLR on path 2 is greater than 3%, the EMS scheme switches to assign more load on path 1. This results in a rather flat in information loss rate at PLR on path 2 greater than 3%.

EMS scheme comes with FEC redundancy and FEC block size adaptations (see Fig. 1). On the other hand, the configuration with Tetrys is simpler than FEC since Tetrys does not need to scale the block size. With the same redundancy ratio, Tetrys achieves smaller information loss rate. Thus, with the redundancy adaptation so that the loss requirement less than a threshold (normally 1% for video), Tetrys requires less redundancy than FEC. In fact, in Fig. 3, the information loss rate of Tetrys is much less than 1% at mean burst size of 3 packets at redundancy ratio of 10% while the redundancy for FEC must be greater than 10% to lower its information loss rate to less than 1%.

Fig. 4 shows the information loss rate of Tetrys and FEC(45,50) at PLR of 3% on both paths and mean burst size of 3 packets. Since Tetrys is fully reliable, the lost packets are due to missed deadline. Thus, the information loss rate of Tetrys is reduced with the relaxation of the delay requirement. This implies that the gain of Tetrys against FEC is increased if the delay constraint is relaxed. It is noted that we use EMS as load splitting scheme to demonstrate the better performance of Tetrys against FEC, we believe that Tetrys still outperform FEC in any load splitting scheme.

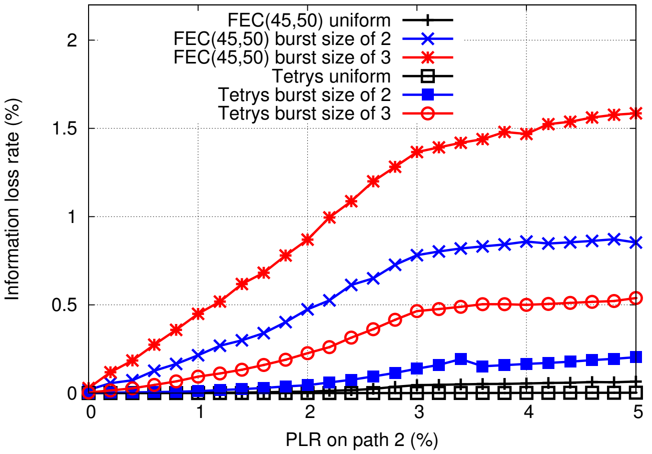

Fig. 3. Tetrys vs FEC(45,50) at PLR of 3% on path 1 with uniform losses and burst losses with mean size of 2 and 3 packets, redundancy ratio is 10%

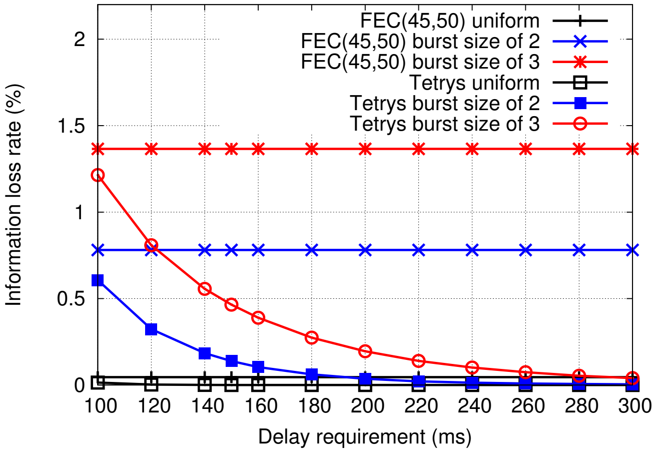

Fig. 4. Information loss rate of Tetrys and FEC(45,50) as a function of delay requirement with PLR of 3% on both paths

B. Propagation delay differences

In [6], the authors did not perform the tests where the available paths have different propagation delays. We compare Tetrys and FEC in case of 3 paths with different settings to IV-A. The PLR on each path is 14%, 10% and 12%, respectively. The redundancy ratio is set to 25% which is equivalent to FEC(15,20), FEC(24,32), FEC(30,40) and FEC(45,60). The simulation duration is 1000s with OLS Adapt Window of 1 second. Kurant showed in [12] that the propagation delay differences between paths reach several tens of milliseconds. Thus, we vary the propagation delay on each path from 50 to 80ms so that no path has the same delay to the others and the maximum delay difference between paths is 30ms. This results in 24 simulations.

First, we compare different strategies of sending Tetrys repair packet (see Table I). Fig. 5 shows the difference in information loss rate of "Tetrys long" against "Tetrys short" and "Tetrys" strategies for the mean burst size of 2 packets. The positive value means that the information loss rate of "Tetrys long" is less than the compared strategy ("Tetrys long" is better) and vice versa. It is clear that "Tetrys long" strategy outperforms other strategies in most cases. Table II shows the mean information loss rate and standard deviation of 24 simulations in case of uniform and mean burst size of 2 and 3 packets. "Tetrys long" strategy shows better results in all cases. At uniform losses and mean burst size of 2, 3 packets, "Tetrys long" gains 50%, 24% and 6%, respectively, against the best strategy among "Tetrys short" and "Tetrys". These results confirm our analyis in III-B. Thus, we consider Tetrys as "Tetrys long" strategy from now on.

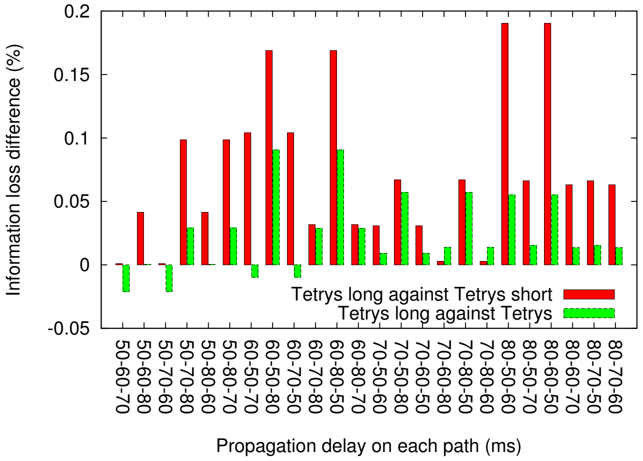

Fig. 5. "Tetrys long" performance against "Tetrys short" and "Tetrys"

TABLE II
MEAN INFORMATION LOSS RATE AND STANDARD DEVIATION OF DIFFERENT STRATEGIES OF SENDING TETRYS REPAIR PACKETS AT UNIFORM LOSSES AND BURST LOSSES WITH MEAN SIZE OF 2 AND 3 PACKETS

	Uniform	Burst size of 2	Burst size of 3
Tetrys long	0.0004% ± 0.00056	0.083% ± 0.021	0.47% ± 0.15
Tetrys short	0.015% ± 0.039	0.15% ± 0.06	0.5% ± 0.1
Tetrys	0.0008% ± 0.0014	0.11% ± 0.04	0.52% ± 0.078

We then compare Tetrys with different FEC settings (FEC(15,20), FEC(24,32), FEC(30,40), FEC(45,60)). Fig. 6 shows the information loss rate of different FEC settings. The larger FEC block size makes FEC more tolerant to burst losses but leads to more delay to recover the lost packets. Fig. 7 shows the comparison between Tetrys and FEC(45,60), the best FEC among 4 settings, at mean burst size of 3 packets. Tetrys outperforms FEC(45,60) regardless the propagation delay on each path.

Table III shows the results of different FEC settings and Tetrys at both uniform losses and burst losses with mean size of 2 and 3 packets. We can see that Tetrys has a significant gain in information loss rate compared to FEC. Specifically, Tetrys has an average gain of 75% in information loss rate against the best FEC at mean burst size of 3 packets.

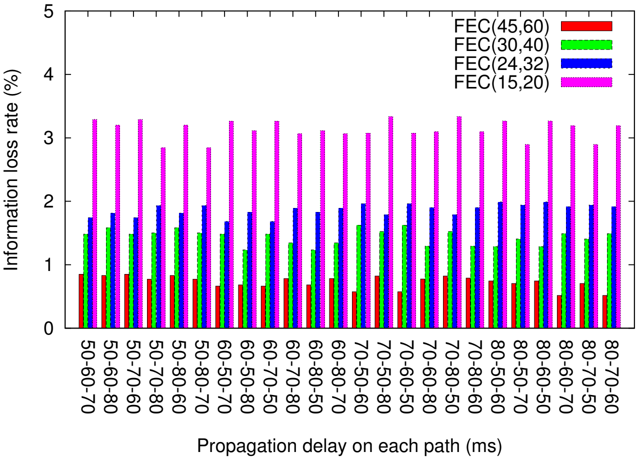

Fig. 6. Comparison between different FEC settings at mean burst size of 2 packets. PLR on each path is 14%, 10%, 12%. Redundancy ratio is 25%

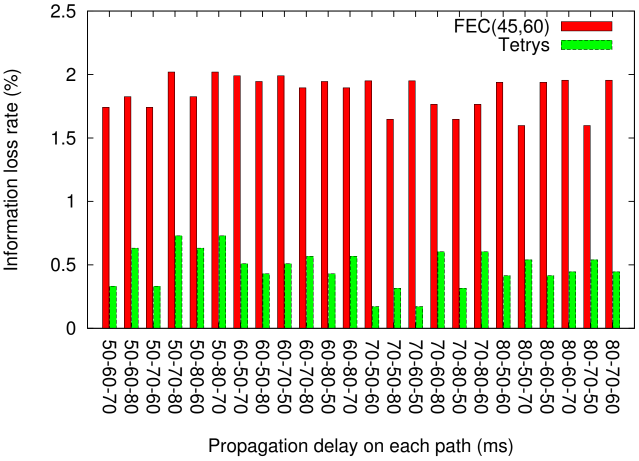

Fig. 7. FEC(45,60) vs Tetrys at mean burst size of 3 packets. PLR on each path is 14%, 10%, 12%. Redundancy ratio is 25%

TABLE III
MEAN AND STANDARD DEVIATION INFORMATION LOSS RATE WITH DIFFERENT FEC SETTINGS AND TETRYS

	Uniform	Burst size of 2	Burst size of 3
FEC(15,20)	0.53% ± 0.14	3.14% ± 0.15	4.77% ± 0.22
FEC(24,32)	0.18% ± 0.051	1.87% ± 0.09	3.55% ± 0.18
FEC(30,40)	0.11% ± 0.041	1.44% ± 0.12	2.81% ± 0.19
FEC(45,60)	0.028% ± 0.017	0.73% ± 0.099	1.86% ± 0.13
Tetrys	0.0004% ± 0.00056	0.083% ± 0.021	0.47% ± 0.15

V. EMS ALGORITHM IMPROVEMENT

The original OLS algorithm (see pseudo code 1) shows very good results. However, it does not adapt well to the network dynamics. In fact, assuming that the OLS is increasing the load on path 1, the loss rate on path 2 reduces significantly and is lower than path 1. This might lead to the better information loss rate, the original OLS scheme is still in a repeat-until loop and continues to increase the load on path 1. This make the OLS scheme goes farther from the new optimal load splitting, while it is better to stop increasing the load on path 1 and to increase the load on path 2. Thus, we propose to add a pre-defined threshold of loss rate θ. At each period, the scheme compares the loss rate on each path with the one in previous period, if the absolute difference is greater than θ, the OLS scheme quits the repeat-until loop and re-sorts the paths. The improved OLS scheme is depicted in the pseudo code 2.

Algorithm 2 Modified OLS

```
1: Compute the asymptotic optimal solution and split the load accordingly
2: Sort the paths in the increasing order of loss rate
3: repeat
4:   Pick the first path in the list
5:   repeat
6:     if the absolute difference of loss rate on one path exceeds the pre-defined threshold $\theta$ then
7:       goto Step 2
8:     end if
9:     Increase the load on the chosen path by pre-defined $\Delta_L$
10:    Decrease the load on each of remaining paths by a fraction of $\delta$, proportional to their respective loss rates
11:  until measured information loss rate increases
12:  Remove the chosen path from the list
13:  Revert to the previous load splitting
14: until the path list is empty
15: goto Step 2
```

With the same settings as in IV-B, we re-run the simulations with a pre-defined threshold $\theta = 5\%$. The information loss rate of both FEC(45,60) and Tetrys with threshold is lower than the one without threshold in case of mean burst size of 2 packets (Fig. 8 and 9). Table IV shows the improvement in information loss rate with modified EMS scheme in both uniform and burst losses. At mean burst size of 2 packets, FEC(45,60) and Tetrys with threshold has an average gain of 30% and 65%, respectively, compared to the case without threshold. While FEC(45,60) and Tetrys with modified EMS scheme have an average gain of 21% and 49%, respectively in comparison to the original EMS scheme at mean burst size of 3 packets. These simulations show that Tetrys achieves much lower information loss rate with modified OLS algorithm although it has a very small information loss rate using the original one.

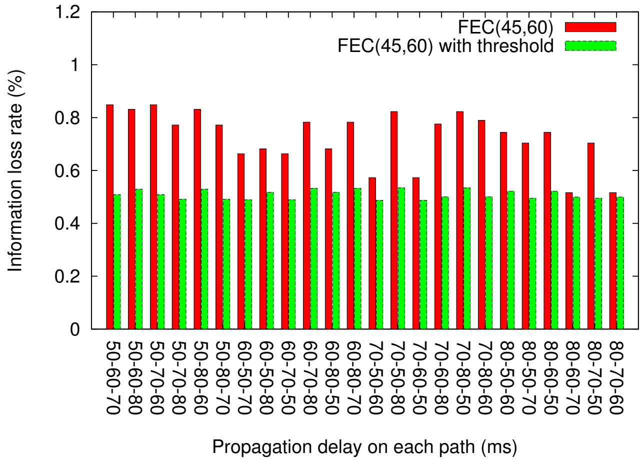

Fig. 8. FEC(45,60) without threshold and with threshold $\theta = 5\%$ at mean burst size of 2 packets

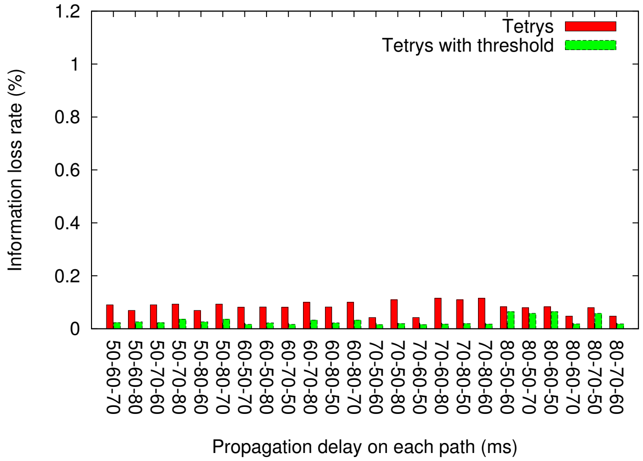

Fig. 9. Tetrys without threshold and with threshold $\theta = 5\%$ at mean burst size of 2 packets

TABLE IV
MEAN AND STANDARD DEVIATION INFORMATION LOSS RATE WITH AND WITHOUT THRESHOLD

		Without threshold	Threshold $\theta = 5\%$
Uniform	FEC(15,20)	0.53% ± 0.14	0.4% ± 0.01
	FEC(24,32)	0.18% ± 0.051	0.12% ± 0.0044
	FEC(30,40)	0.11% ± 0.041	0.056% ± 0.0053
	FEC(45,60)	0.028% ± 0.017	0.014% ± 0.0058
	Tetrys	0.0004% ± 0.00056	0.00016% ± 0.00025
Burst size of 2	FEC(15,20)	3.14% ± 0.15	2.71% ± 0.019
	FEC(24,32)	1.87% ± 0.09	1.58% ± 0.02
	FEC(30,40)	1.44% ± 0.12	1.12% ± 0.014
	FEC(45,60)	0.73% ± 0.099	0.51% ± 0.017
	Tetrys	0.083% ± 0.021	0.029% ± 0.016
Burst size of 3	FEC(15,20)	4.77% ± 0.22	4.45% ± 0.024
	FEC(24,32)	3.55% ± 0.18	3.07% ± 0.022
	FEC(30,40)	2.81% ± 0.19	2.44% ± 0.038
	FEC(45,60)	1.86% ± 0.13	1.47% ± 0.036
	Tetrys	0.47% ± 0.15	0.24% ± 0.071

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced an on-the-fly coding scheme named Tetrys to real-time multipath streaming. With the same load splitting scheme, the EMS scheme presented in [6], we have shown that Tetrys consistently has significant reduction in information loss rate compared to the FEC approach in both uniform and burst losses. We showed that the decoupling between load allocation and Tetrys redundancy traffic improves the performance in terms of loss rate after packet recovery. The Tetrys repair packets are preferably sent to the path with longer propagation delay shows best performance. Furthermore, we showed that the EMS scheme can be improved to provide better results. By introducing a threshold parameter, the modified EMS scheme adapts well to the network dynamics and showed a significant reduction in information loss rate compared to the original one. For future work, we plan to analyze the multipath streaming in more realistic contexts (e.g., 2 paths with Wi-Fi and 3G/LTE) and to validate the results with video data.

VII. ACKNOWLEDGMENTS

This work was supported by the French ANR grant ANR-VERS-019-02 (ARSSO project).

REFERENCES

[1] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan, "Best-path vs. multi-path overlay routing," in *IN PROC. ACM SIGCOMM Internet Measurement Conference*, 2003, pp. 91–100.

[2] L. Golubchik, J. C. Lui, T. F. Tung, A. L. Chow, W. j. Lee, G. Franceschinis, and C. Anglano, "Multi-path continuous media streaming: What are the benefits?" 2002.

[3] S. Fashandi, S. O. Gharan, and A. K. Khandani, "Path diversity over packet switched networks: performance analysis and rate allocation," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 1373–1386, October 2010.

[4] D. Jurca and P. Frossard, "Media flow rate allocation in multipath networks," *IEEE Transactions on Multimedia*, p. 12271240, 2007.

[5] D. Jurca, P. Frossard, and A. Jovanovic, "Forward error correction for multipath media streaming," *IEEE Trans. Circuits and Systems for Video Technology*, pp. 1315–1326, 2009.

[6] A. L. H. Chow, H. Yang, C. H. Xia, M. Kim, Z. Liu, and H. Lei, "EMS: Encoded Multipath Streaming for Real-time Live Streaming Applications," in *IEEE International Conference on Network Protocols, Princeton, NJ, USA, 13-16 October 2009*. IEEE Computer Society, 2009, pp. 233–243.

[7] T. Nguyen and A. Zakhor, "Path Diversity with Forward Error Correction (PDF) System for Packet Switched Networks," in *in Proceedings of IEEE INFOCOM*, 2003, pp. 663–672.

[8] J. Kumar Sundararajan, D. Shah, and M. Medard, "ARQ for network coding," in *IEEE International Symposium on Information Theory*, july 2008, pp. 1651 –1655.

[9] J. Lacan and E. Lochin, "Rethinking reliability for long-delay networks," in *Proceedings of International Workshop on Satellite and Space Communications IWSSC*, Oct. 2008.

[10] P.-U. Tournoux, E. Lochin, J. Lacan, A. Bouabdallah, and V. Roca, "On-the-fly coding for time-constrained applications," *IEEE Transactions on Multimedia*, vol. 13, 2011.

[11] G. Cheung, "Near-optimal multipath streaming of H.264 using reference frame selection," in *Proceedings of International Conference on Image Processing*, sept. 2003.

[12] M. Kurant, "Exploiting the path propagation time differences in multipath transmission with fec," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 5, pp. 1021–1031, 2011.

[13] "The Network Simulator http://www.isi.edu/nsnam/ns/index.html."

[14] "ITU-T recommendation G.114," International Telecommunication Union, Tech. Rep., 2009.

[15] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 645–656, 2003.