Capillary multi-path routing with FEC for real-time multimedia

Emin Gabrielyan
EPFL and Switzernet Sàrl
Lausanne, Switzerland
emin.gabrielyan@{epfl.ch,switzernet.com}

Abstract – **Use of FEC in off-line streaming, offering large buffers, gives spectacular results; but real-time streaming puts hard restrictions on the buffer size leaving FEC helpless for combating long link failures on a single path route. Another orthogonal method, the multi-path routing, can make FEC effective also for the real-time streaming, which has been already shown on examples with an alternate path. In this paper we introduce capillary routing algorithm offering a wide range of multi-path routing topologies starting from a simple (max-flow multi-path) solution toward more reliable and secure schemes obtained by recursively spreading individual sub-flows. The friendliness of a particular multi-path routing is measured by Adaptive Redundancy Overall Need (ARON), which is proportional to the sender's total channel coding effort needed for combating the failure of each link in the multi-path route. A dozen of capillary routing layers, built on several hundreds of network samples obtained from a random walk wireless Mobile Ad-Hoc Network (MANET), are rated with ARON showing that the FEC friendliness improves substantially as the routing grows more spreader.**

I. Introduction

Packetized IP communications behave like erasure channels. Data is chopped into packets, and since each packet is either received without error or not received, erasure resilient FEC codes, applied at the packet level, can mitigate packet losses.

In off-line packetized applications FEC offered spectacular results. Via satellite broadcast channel with erasure resilient Raptor codes [1] it is possible to recurrently update voluminous GPS maps to millions of motor vehicles under conditions of arbitrary fragmental visibility and without a feed-back channel. In the film industry LT code [2] enables a fast delivery over the lossy internet of the day's film footage from the location it has been shot to the studio that is many thousands of miles away (impossible with TCP, converging the bandwidth, irrespectively to the effective capacity, to a function from the packet loss rate and RTT, unavoidably high due to distance). 3GPP, 3rd Generation Partnership Project, recently adopted Raptor as a mandatory code in Multimedia Broadcast/Multicast Service (MBMS), for its significant performance in file transfer [3], [4], [5].

An important reason the above examples of off-line streaming can significantly benefit from FEC is that, contrary to the real-time streaming, the application is not obliged to deliver in time the "fresh" packets of a very short life time and the buffer size is not a concern. When buffer size is restricted, FEC can only mitigate short granular failures. Many studies reported weak or negligible improvements from application of FEC to real-time streaming. In [6] it has been shown improvements from the application of FEC only if the stochastic packet losses range is between 1% and 5%. For real-time packetized streaming the author of [7] proposed to combine FEC with retransmissions. In [8] it has been reported a high overhead from the use of FEC during bursts. The author of [9] claims that for two-way, delay-sensitive real-time communications, the application of FEC on the packet level can not give any valuable results at all.

Studies stressing on the poor FEC efficiency always assumed that the media stream follows a single path. Exploiting other dimensions which can "replace" the long buffering time can nevertheless make FEC significantly efficient in the fault-tolerance of real-time streaming. The underlying network routing is an ax, orthogonal to the buffering, which should be exploited. There is an emerging body of a literature addressing the path diversity for improving the efficiency of FEC [10], [11], [12], [13], [14] and [15], but the diversity in these studies is limited to either two (possibly correlated) paths or in the best case to a sequence of parallel and serial links. Converting a single path routing to the basic multi-path routing, however, is not the terminal achievement of the multi-path approach. The routing topologies, so far, were not regarded as a space of possibilities and a ground for searching a FEC effective pattern.

In this paper we try to present a comparative study across a range of friendly, all multi-path routing patterns virtually erected along a routing ax. Single path routing, being considered as too hostile, will be even excluded from our comparison system.

As an approach to multi-path routing concept we propose a family of *capillary routing*. In capillary routing, the alternate paths are discovered by delegating the load of a single path route to other links. The load balance is reached by minimizing the upper bound value of the flow for all links. Capillary routing is built layer by layer, providing at each layer a multi-path routing suggestion, growing spreader as the layer number augments. The first layer is the simplest multi-path routing representing a max-flow solution. Successive layers are obtained by recursively spreading the individual sub-flows of previous layers. With capillarization of the routing, the communication uses the network more reliably penetrating into the full outfit of transmission capacities offered by the network topology. The last layer represents the complete capillary routing and, contrary to the shortest path or max-flow has one unique solution for any given network and pair of peers. We present the capillary routing construction in section II.

To compare two multi-path routing suggestions, we are introducing a measure of the routing's advantageousness based on the satisfaction level of a realistic application employing end-to-end adaptive FEC. Adjustable FEC in real-time streaming was already proposed for implementation in practice by several other authors [16], [17], [8], [6] and [7]. The end-to-end adaptive FEC mechanism is not aware of the underlying routing and is implemented entirely in the application level of end nodes. Sender is streaming the media equipped by default with a constant FEC tolerating a certain packet loss rate. Packet loss rate is measured at the receiver and is constantly reported back to the sender. The sender must always increase the FEC overhead whenever the packet loss rate is about to exceed the tolerable limit. We use Adaptive Redundancy Overall Need (ARON), the total amount of the adaptive redundancy demanded from the sender during the communication time, as a measure of the cooperativeness and friendliness of the underlying network routing toward FEC. The novelty brought by ARON, which is introduced in details in section III, is that a routing topology of any complexity can be rated by a single scalar value.

Further, in section IV, we evaluate the advantageousness of the capillarization by rating each layer of capillary routing with ARON. Network samples for the measurements are obtained from a wireless random walk Mobile Ad-hoc Network (MANET) with hundreds of nodes. Our study has shown that in scope of multi-path routing, significant improvement can be still obtained by improving the basic path diversity provided by the first layer of the capillary routing (i.e. the max-flow solution) toward more elaborated multi-path routing strategies provided by the deeper layers of the capillary routing. Multi-path routing ax alone, similarly to the buffering ax, can substantially burst the efficiency of FEC.

II. Capillary routing

Capillary routing seeks to minimize the impact of individual link failures to the media stream giving thus the encoder a greater chance to recover the failure.

The strategy can be best defined by describing an iterative LP process transforming a simple single-path flow into a capillary route. First minimize the maximal value of the load of links by minimizing an upper bound value applied to all links. By balancing so the load of all links, in the first layer, the full mass of the flow is split equally across the available parallel routes. Further, find the bottleneck links of the first layer. Maintaining the first upper bound (applied to all links) on its minimal level, minimize the maximal load of remaining links by minimizing another upper bound value applied to all links except the previous bottleneck links. The objective of the second iteration discovers the sub-routes and the sub-bottlenecks of the second layer. Find the bottleneck links of the second layer. Minimize the maximal load of remaining links, now without also the bottlenecks of the second layer, and continue the iteration until the entire footprint of the flow is discovered. A flow traversing a large dense network with hundreds of nodes may have hundreds of capillary routing layers.

The next figures show three layers of the capillary routing on a network example with 7 nodes (from Fig. 1 thru Fig. 3). The top node on the diagrams is the sender and the bottom node is the receiver; all links have a direction from up to down.

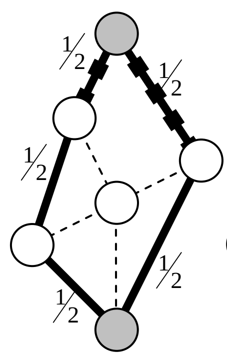

Fig. 1. In the first layer the flow is equally split over two paths, two links of which, marked by thick dashes, are the bottlenecks.

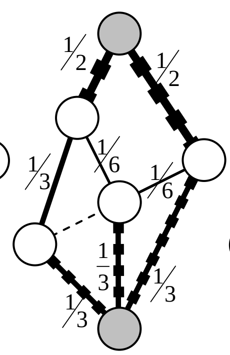

Fig. 2. The second layer minimizes the maximal load of the remaining seven links and finds three bottlenecks, each with a load of 1/3.

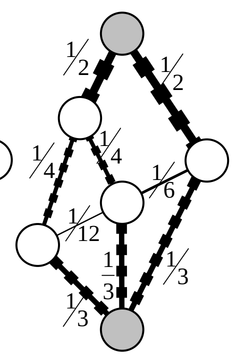

Fig. 3. The third layer minimizes the maximal load of the remaining four links and finds two bottlenecks each with a load of 1/4.

In the first layer of the capillary routing (Fig. 1) there are four links having a load of $\frac{1}{2}$, but only two of them, marked by thick dashes, are the true bottlenecks and will continue to maintain their value. In the second layer (Fig. 2) there are four links with a value of $\frac{1}{3}$, but only three of them, also marked in thick dashes, are the true bottlenecks. In the final third layer (Fig. 3) there are two bottlenecks with a load of $\frac{1}{4}$. As for the remaining two links with loads of $\frac{1}{6}$ and $\frac{1}{12}$, there is no freedom left for them.

Although the LP approach derived straight from the definition of the capillary routing is fully valid; precision errors propagating through the sequence of LP minimizing problems can often reach noticeable sizes and sometimes, when reaching tiny loads, can even result in infeasible LP problems. We have found thus a different, stable LP method maintaining the parameters and variables always in the same order of grandeur.

Instead of decreasing the maximal value of loads of links, the routing path is discovered by solving the max flow problem. The resulting paths of these two methods are identical except that the proportions of flow are different by the increase factor of the max flow solution. Te below diagrams of Fig. 4 thru Fig. 9 present the capillary routing discovery with the max-flow LP approach, on the same example with 7 nodes previously shown in Fig. 1, Fig. 2 and Fig. 3.

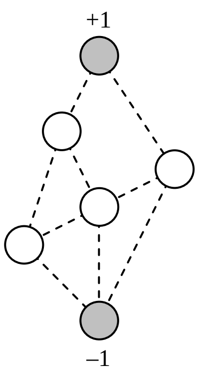

Fig. 4. Initial problem with one source and one sink node

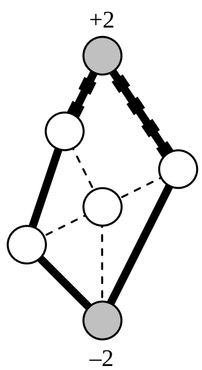

Fig. 5. Maximize the flow, fix new flow-out coefficients at the nodes, find the bottleneck links

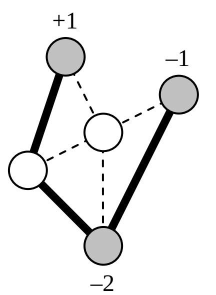

Fig. 6. Remove the bottleneck links from the network adjusting the flow-out coefficients at the adjacent nodes

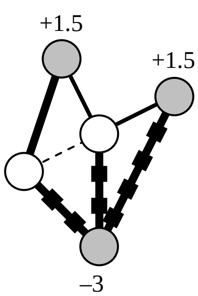

Fig. 7. Maximize the flow in the obtained new problem, fix new flow-out coefficients at the nodes, find the new bottlenecks

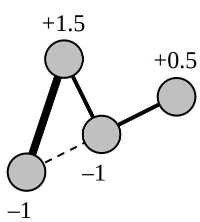

Fig. 8. Remove the bottleneck links from the network adjusting correspondingly the flow-out coefficients at the adjacent nodes

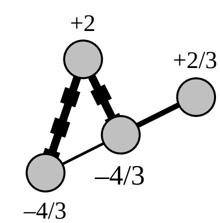

Fig. 9. Again maximize the flow in the obtained new problem and fix the new resulting flow-out coefficients at the nodes

The max-flow problem is defined by the flow-out coefficients at each node. Initially only the peer nodes have non-zero flow-out coefficients, +1 for the source and –1 for the sink. Generally, at each construction layer we have a synchronous multiple-multicast problem: a uniform flow from set of sources to set of sinks, where all rates of transmissions by sources and all rates of receptions by sinks are subject of proportional (i.e. synchronous) increase respecting each node's flow-out coefficient (either positive or negative).

The LP problem of the successive layer is obtained by complete removal of the bottlenecks from the problem, adjusting correspondingly the flow-out coefficients of the adjacent nodes (to respect the flow conservation rule) and thus possibly producing new sources and sinks in the network. The successive layers in general, except the unicast problem of the first layer, do not belong to the simple class of "network linear programs" [18]. Presenting the iteration of layers in equations:

Let the flow problem of a synchronous multiple-multicast at the layer *l* is defined as follows:

- set of nodes N^l,
- set of links $(i,j) \in L^l$, where $i \in N^l$ and $j \in N^l$,
- and flow-out values f_i^l for all $i \in N^l$

And let in its max-flow solution:

- the synchronous flow increase factor for all nodes is F^l
- and the set of bottlenecks is B^l (where $B^l \subset L^l$)

Then the new problem of the successive layer: N^{l+1}, L^{l+1} and f^{l+1} is defined as follows:

- $N^{l+1} = N^l$
- $L^{l+1} = L^l - B^l$
- $f_j^{l+1} = f_j^l \cdot F^l + \sum_{(i,j)\in B^l}(+1) + \sum_{(j,k)\in B^l}(-1)$

After a certain number of applications of the max-flow with corresponding modifications of the problem, we will finally obtain a network having no source and sink nodes. At this moment the iteration stops. All links followed by the flow in the capillary routing will be enclosed in bottlenecks of one of the layers.

The absolute flow $r_{i,j}$ traversing the link $(i,j) \in B^l$ in the capillary routing must be computed according the following equation:

$$r_{i,j} = \frac{1}{\prod_{i=1}^{l} F^i}$$, where *l* is the layer for which $(i,j) \in B^l$

The max-flow approach proved to be very stable, because it maintains all values of variables and parameters within a close range of 1 (even for very deep layers with tiny loads) and also because it enables to re-calibrate the LP problem before parsing to the next layer avoiding thus the undesirable propagation of precision errors (calibration details are out of scope of this paper).

Description of the bottleneck identification was deliberately left for the end of this section, since behind this phase is hidden a repeated application of another LP model. Bottlenecks of each max-flow solution are discovered in a bottleneck hunting loop. Each iteration of the hunting loop is an LP cost minimizing problem that reduces the load of traffic over all links being suspected as bottlenecks. Only links maintaining their load on the initial level will be passed to the next iteration. Links undergoing to load reduction under the LP objective are removed from the suspect list and the bottleneck hunting iteration stops if there are no more links to remove.

III. Adaptive Redundancy Overall Need (ARON)

Spread routing alone would not solve the problem of tolerance without FEC. If media stream is not capable to sustain any losses at all, by spreading the transmission the media becomes even more vulnerable, since there are more links whose failures will damage the stream. Most of real-time media streaming applications are tolerant to a certain level of packet losses due to passive error concealment or media encoding techniques (e.g. a packet may carry duplicates of media from previous slots, encoded with a lower rate source coding, etc). VOIP for example can tolerate 8% to 11% packet losses. The static tolerance can be also obtained or increased by a constant FEC code. We propose to combine the little static tolerance of the media stream, combating weak failures, with a dynamically added adaptive FEC, combating the strong failures exceeding the tolerable packet loss rate.

ARON is defined over all links as the sum of FEC efforts, each of which is the percentage by which the sender must increase its packet transmission rate to combat the individual link failure of the route. For example if the communication footprint consists of five links and in response to each link failure the sender increases the packet transmission rate by 25%, then ARON will be equal to the sum of these five FEC efforts, i.e. to 1.25.

Let P be the usual packet transmission rate and P_l be the increased rate of the sender, responding to the failure of the link $l \in L$, where L is the set of all links. Then according to the definition:

$$ARON = \sum_{l \in L} \left(P_l / P - 1 \right)$$

Let us have a long communication, and let D be the total failure time of a single network link during the whole duration of the communication (which is the product of the duration of a single link failure, the frequency of a single link failure and the total time of the communication). Then $D \cdot P \cdot ARON$ is the number of extra redundant packets that the sender will transmit in order to compensate all arbitrary network failures occurring during the whole communication time (assuming a single link failure at a time and a uniform probability and duration of link failures). Parameters D and P are constants in respect to various routing suggestions between the peers, but ARON is the component carrying indications on the size and topology of the particular multi-path routing. In other words, for the total number of extra redundant packets or for the cost of the redundancy in communication, ARON is the proportionality coefficient representing solely and fully the routing.

Redundant packets are injected in the original stream of media slots for every block of M media packets using a systematic erasure resilient code (thus without a need to affect the original media packets). During the streaming, the length of the block of media packets (M), limited by the receiver's playback buffer time, is supposed to stay constant. Number of redundant packets for each block of M media packets is however a variable, depending on the conditions of the erasure channel. The M media packets with all related redundant packets form a FEC block. By FEC_p we denote the FEC block size chosen by the sender in response to the packet loss rate p. We are assuming that the media stream obtains the static tolerance to losses $0 \le t < 1$ by maintaining constant FEC code, which by default streams the packets in FEC blocks of the length of FEC_t. Only if the loss rate p measured at the receiver is about to exceed the tolerable limit t the sender must increase its transmission rate by injecting additional redundant packets.

The random packet loss rate, observed at the receiver during the failure (or congestion) time of a link in the communication path, is the portion of the traffic being still routed toward the faulty route. Thus a complete failure of a link l carrying according to the routing pattern a load of traffic $0 \le r(l) \le 1$ will produce at the receiver a random packet loss rate equal to the same $r(l)$. The equation for ARON can thus be re-written as follows:

$$ARON = \sum_{l \in L \,|\, t \le r(l) < 1} \frac{FEC_{r(l)} - FEC_t}{FEC_t}$$

The critical links of the path carrying the entire traffic are ignored, since the FEC required for the compensation of long failures of such links would be infinite. All multi-path routing schemes that are subject of comparisons in scope of our study are smart enough to delegate the load from a critical route over other links, if of course the network topology makes it possible. Thus our comparisons cannot contain a really bad routing sample following a single path route on a link when alternate multiple path routes are possible; and if the link is really critical by the network topology, then without a risk of affecting the comparison results such a link can be removed from all suggested routings in order to compare their remaining portions (since any routing unavoidably will pass its entire traffic through the truly critical link). We do not therefore demonstrate the advantage of the multiple path strategies versus single path routing.

We compute FEC_p function assuming a Maximum Distance Separable (MDS), e.g. Reed-Solomon code. By the choice of an MDS code, the reception of the sufficient number of any type of transmitted packets, precisely said, exactly the same number as there were media packets in the block, is the only condition for the successful decoding of all original media packets.

A parameter we must rely on for computing the transmission block size $FEC_p \ge M$ is the Decoding Error Rate (DER) – the desired decoding failure probability at the receiver.

To collect mean M packets at the receiver we must transmit $M/_{1-p}$ packets at the sender. However the probability of receiving $M-1$ packets or $M-2$ packets (which makes the decoding impossible) is quite high. Therefore for small values of M, which is the limiting factor of the real-time media, we must send much more redundant packets in the block than is necessary to receive mean M packets on the receiver side. The mean of received packets should be maintained much higher than M and the probability of receiving less than M packets must be maintained very low, below DER.

The probability of having exactly n losses (erasures) in a block of N packets with a random loss probability p is computed by binomial distribution:

$\binom{N}{n} \cdot p^n \cdot q^{N-n}$, where $\binom{N}{n} = \frac{N!}{n! \cdot (N-n)!}$ and $q = p - 1$

The probability of having $N - M + 1$ and more losses (i.e. less than M survived packets), is computed as follows:

$$\sum_{n=N-M+1}^{N} \binom{N}{n} \cdot p^n \cdot q^{N-n}$$

The above formula gives the decoding failure probability if the FEC block size is equal to N. Therefore for computing the carrier block's minimally needed length for a satisfactory communication, it is sufficient to steadily increase the carrier block length N until the desired decoding error rate (DER) is met.

With quick search methods and efficient methods for computing values of the binomial distribution, especially fast when the value at the neighboring position is already computed, we can quickly build the FEC block size integer function from the random loss probability $0 \le p < 1$. Further, a very coarse approximation of the failure probability enabled us also to find the right function for interpolation of the discrete integer values into a smooth FEC_p function (details of which are out of scope of the present paper).

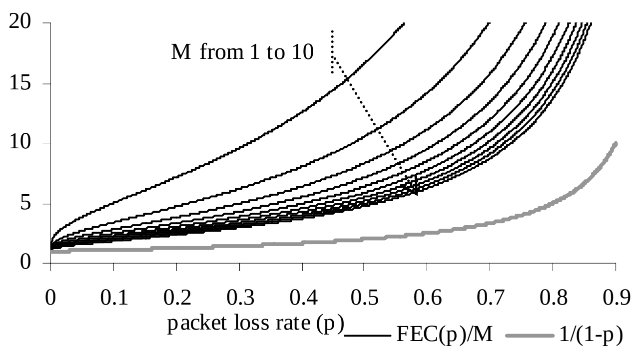

Fig. 10. FEC overhead as a function from the packet loss rate

Several FEC_p / M functions (FEC overheads) for media packets from 1 to 10 are plotted in the chart of Fig. 10 (for $DER = 10^{-5}$). These functions are compared with $1/_{1-p}$, derived from the Shannon capacity. Higher is the number of media packets in the block (i.e. longer is the buffering time) closer the communication can approach to Shannon conditions.

In real-time streaming there is a hard tradeoff between M and the cost of FEC overhead. Before playing the media, the receiver must hold in the buffer enough packets to restore the recoverable losses. The receiving side of the media application is already equipped with a playback buffer to compensate the network jitter and to reorder packets arriving in wrong order. The playback buffer must be extended enough to consider also holding of packets of the FEC block (at least M packets). Despite many arguments in favor of long M, for example in VOIP with 20 ms sampling rate (g729r8 or AMR codec) the number of media packets in a single FEC block must not exceed 20 – 25 packets.

M can be very long in off-line applications such as file transfer. An example of nearly off-line application for which the path diversity can be essential is the problem of the last kilometer bottleneck for an internet user downloading and watching a one-way video from multiple servers (studied by [14], [20] and [21]). The buffering time can be few minutes long with thousands of media packets in a single FEC block. Capacity approaching fountain codes [19] are the best for this kind of application. Path diversity is needed, because we desire to maintain the reception at the constant rate of the maximal bandwidth of the last kilometer. To ensure the reception at the maximal downlink rate through a lossy network the sender must transmit at a variable rate combating the arbitrary losses arising in different network locations.

For this particular application assuming near optimal Shannon condition $FEC_p = M/_{1-p}$, and the following equation of ARON can be derived:

$$ARON = \sum_{l \in L \mid t \le r(l) < 1} \left(\frac{1-t}{1-r(l)} - 1 \right)$$

The next section presents rating of various routing suggestions for real-time media with short buffering time.

IV. Measuring the Friendliness of Capillary Routing

We need to compute average ARON on various network samples to evaluate the overall performance of the capillary approach. First we suggest the first layer routing individually to each network sample and we obtain thus the average ARON rating for all routing (max-flow) suggestions of the first layer. Then we compute the second layer routing individually for each network sample in the same set and we obtain therefore the set's average ARON rating for the routing suggestions of the second layer. We measure similarly the average ARON for the capillary routing layers from 1 to 10 on the same set of network samples obtaining thus an overall figure of the performance as the layer number grows.

In Fig. 11 below we have seven sets, each containing 42 network samples. At the same time we consider also 15 media streams different by their static tolerance to losses varying from 3.6% to 7.8%. Thus for each set we have 15 curves of average ARON and all of them decrease as the capillary routing layer increases from 1 thru 10 demonstrating the improvements delivered by higher layers. Increase of the capillary routing layer, i.e. spreading of the basic multi-path routing also through the non-bottleneck portions of the network, reduces the ARON sensibly and therefore also the FEC effort of the sender combating the link failures and packet losses.

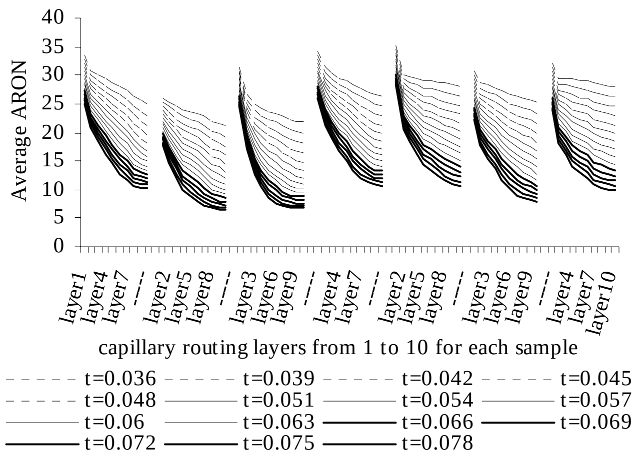

Fig. 11. Average ARON as a function from the capillary routing layer

Logically, the ARON curve of the media stream is shifted down by every unit of the statically added tolerance. At the same time it is interesting to observe that the presence of a little static tolerance in the media stream stresses the efficiency gain achieved by the deeper layers much stronger than for streaming examples with weak static tolerance. Although there are hundreds layers in the complete capillary routing, the first few layers alone reduce the average FEC effort of the sender by a factor of three. According to the chart the gain from additional spreading is insignificant after the layer 8 or 9.

Of course, the exact pattern of the ARON improvement curve, as a function from the layer, depends on the distance between the peers, the network size and its density. The network samples for the above chart are obtained from a random walk wireless Mobile Ad-hoc Network (MANET). Initially the nodes are randomly distributed on the rectangular area, and then at every timeframe they move according to a random walk algorithm. If two nodes are close enough (and are within the coverage range) then there is a link between them. In the above example, there are 115 nodes and 300 timeframes each leading to a different network sample (all of which are broken into 7 sets represented on the above chart). The number of media packets per each transmission block (M) is 20 and the desired decoding failure rate (DER) is 10^{-5}.

Multi-path routing suggestions for fault-tolerant streaming are applicable not only to Ad-Hoc or sensor networks, but also to mobile networks, where wireless content can be streamed to and from the user via multiple base stations; or to the public internet, where, if the physical routing cannot be improved, a near capillary routing can be still obtained through an overlay network using peer-to-peer relay nodes or media routers (for suggestions of relay nodes see [15]).

We hope that our investigation will provide some guidelines for future design of diversity-based multimedia transmission systems.

As for an immediate use, fault-tolerant media streaming over a public internet can relay on a network of cheap media routers content unwarily redirecting the UDP traffic. An Internet Telephony Service Provider (ITSP), may collocate or host hundreds of peer-to-peer relay nodes in various network locations (especially beneficiary is the hosting in premises of those ISPs who connect a noticeable numbers of the clientele of ITSP). Spreading of the flow from the user agents (UA) to the media routers can be implemented in the firmware of a SIP phone, in the NAT router of the user or in the closest SIP proxy.

Spreading can be obtained more transparently at a lower network layer using VPN tunnels such that the flow at the source is split across VPN interfaces each leading to various VPN gateways scattered across the network. Alternatively, limited path diversity can be obtained also by assigning to media gateways or SIP servers more than one IP interfaces, each obtained from a different ISP.

Since capillary routing is independent from streaming, an ISP, wishing to be media friendly, can capillarize the routing of its network for the entire mix of UDP packets, does not matter if they carry, high or low resolution video or voice. Most IP routers, including all recent IOS releases of Cisco, provide load balancing in static routing or in EIGRP mode; the last must be however used in the packet load balance mode and not in the session mode.

V. Conclusion

We introduce a multi-path capillary routing, which is built layer by layer. The first layer provides a simple max-flow solution, but as the layer number augments the underlying routing, growing spreader, relays on the network more securely. We also introduced ARON, a method for rating a multi-path routing by a single scalar value. ARON corresponds to the total encoding effort the sending node needs to provide for combating the losses occurring from the (non-overlapping) failures of links in the communication path. By rating the friendliness of the layers of the capillary routing, we have shown that by improvements of the routing topology we can increase substantially the FEC efficiency of multi-path streaming.

VI. REFERENCES

[1] Amin Shokrollahi, “Raptor codes”, ISIT’04, June 27 – July 2, page 36

[2] Michael Luby, “LT codes”, FOCS’02, November 16-19, pp. 271-280

[3] Technical Specification Group Services and System Aspects, “Report of FEC selection for MBMS”, Mar 14-17, 2005, http://www.3gpp.org

[4] Technical Specification Group Services and System Aspects, “Efficient FEC code for reliable MBMS user services”, Mar 14-17, 2005, http://www.3gpp.org

[5] Technical Specification Group Services and System Aspects, “Report of MBMS FEC Status”, Jun 6-9, 2005, http://www.3gpp.org

[6] Ingemar Johansson, Tomas Frankkila, Per Synnergren, “Bandwidth efficient AMR operation for VoIP”, Speech Coding 2002, Oct 6-9, pp. 150-152

[7] Yicheng Huang, Jari Korhonen, Ye Wang, “Optimization of Source and Channel Coding for Voice Over IP”, ICME’05, Jul 06, pp. 173-176

[8] Chinmay Padhye, Kenneth J. Christensen, Wilfrido Moreno, “A new adaptive FEC loss control algorithm for voice over IP applications”, IPCCC’00, Feb 20-22, pp. 307-313

[9] Eitan Altman, Chadi Barakat, Victor M. Ramos, “Queueing analysis of simple FEC schemes for IP telephony”, INFOCOM 2001, Vol. 2, Ap 22-26, pp. 796-804

[10] Qi Qu, Ivan V. Bajic, Xusheng Tian, James W. Modestino, “On the effects of path correlation in multi-path video communications using FEC over lossy packet networks”, IEEE GLOBECOM’04 Vol. 2, Nov 29 - Dec 3, pp. 977-981

[11] Tawan Thongpook, “Load balancing of adaptive zone routing in ad hoc networks”, TENCON 2004, Vol. B, Nov 21-24, pp. 672-675

[12] Rui Ma, Jacek Ilow, “Reliable multipath routing with fixed delays in MANET using regenerating nodes”, LCN’03, Oct 20-24, pp. 719-725

[13] Rui Ma, Jacek Ilow, “Regenerating nodes for real-time transmissions in multi-hop wireless networks”, LCN’04, Nov 16-18, pp. 378-384

[14] Thinh Nguyen, Avideh Zakhor, “Protocols for distributed video streaming”, Image Processing 2002, Vol. 3, Jun 24-28, pp. 185-188

[15] Thinh Nguyen, P. Mehra, Avideh Zakhor, “Path diversity and bandwidth allocation for multimedia streaming”, ICME’03 Vol. 1, Jul 6-9, pp. 663-672

[16] Seong-ryong Kang, Dmitri Loguinov, “Impact of FEC overhead on scalable video streaming”, NOSSDAV’05, Jun 12-14, pp. 123-128

[17] Youshi Xu, Tingting Zhang, “An adaptive redundancy technique for wireless indoor multicasting”, ISCC 2000, Jul 3-6, pp. 607-614

[18] Robert Fourer, *A modeling language for mathematical programming*, Thomson – Brooks/Cole, second edition, 2003, page 343

[19] David J. C. MacKay, “Fountain codes”, IEE Communications, Vol. 152 Issue 6, Dec 2005, pp. 1062-1068

[20] John W. Byers, Michael Luby, Michale Mitzenmacher, “Accessing multiple mirror sites in parallel: using Tornado codes to speed up downloads”, INFOCOM 1999, Vol. 1, Mar 21-25, pp. 275-283

[21] Tuna Guven, Chris Kommareddy, Richard J. La, Mark A. Shayman, Bobby Bhattacharjee “Measurement based optimal multi-path routing”, INFOCOM 2004, Vol. 1, Mar 7-11, pp. 187 - 196