Spread routing with FEC for real-time multimedia in wireless networks

Emin Gabrielyan
École Polytechnique Fédérale de Lausanne and Switzernet Sàrl
emin.gabrielyan@{epfl.ch,switzernet.com}

Abstract

Efficiency of Forward Error Correction (FEC) in packet switched networks strongly depends on the buffer size. 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. Apart the buffering ax, there exists however another orthogonal method, the multi-path routing, which can make FEC effective also for the real-time streaming. Previous works already demonstrated the advantage of the path diversity but limited themselves to restricted topologies, often double-path examples. Although the pure fact of diversity is an important step, we, however, explore also the wide space of possible multi-path routing topologies. For measuring the friendliness of a particular multi-path routing suggestion, we introduce Adaptive Redundancy Overall Need (ARON), which is proportional to the total encoding effort needed for combating the failure of each link in the multi-path route. A novelty brought by ARON is that a routing topology of any complexity can be rated by a single scalar value. Finally, we introduce capillary routing iterative algorithm leading us to a large range of multi-path spread routing patterns, starting from a simple (max-flow multi-path) suggestion toward more elaborated and spreader solutions (by individually equilibrating the sub-flows of previous solutions). Capillary routing layers are built on numerous network samples obtained from a random walk wireless Mobile Ad-Hoc Network (MANET) and we rate them with ARON showing that the FEC friendliness improves substantially as the routing grows more spreader.

1. Introduction

Packetized IP communications over error prone wireless network behave like erasure channels. Files or packetized media are chopped into packets, and each packet is either received without error or not received. To combat packet losses erasure resilient FEC codes can be applied on the packet level.

For numerous packetized off-line applications, employment of erasure resilient codes offered spectacular results. Wireless recurrent updates of voluminous GPS maps to millions of motor vehicles under the conditions of arbitrary fragmental visibility is possible via feedback-less satellite broadcast channel, thanks to the erasure resilient Raptor code [1]. LT codes [2] is used in the film industry for a fast delivery over Internet of the day's film footage from the location it has been shot to the studio that is many thousand miles away. 3GPP, 3rd Generation Partnership Project, recently adopted Raptor as a mandatory code in Multimedia Broadcast/Multicast Service (MBMS), for its significant performance in media broadcasting and file transfer [3], [4], [5].

An important reason the above examples of off-line streaming can significantly benefit from application of 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 long buffering 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. Only 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 other ax, orthogonal to that of the buffering, which should be exploited, is the underlying network routing. There is an emerging body of a literature addressing the advantages of multi-path routing in media streaming and suggesting the important and promising impact the routing can have to the efficiency of FEC [10], [11], [12], [13], [14] and [15]. These studies, however, prove only the advantageousness of path diversity versus single path routing. The considered topologies and routing patterns in these papers are simple and are limited to only two alternate paths or in the best case to a sequence of parallel and serial links. The first step in the path diversity, that is converting a single path routing to the basic multi-path routing, is not however the terminal achievement of the multi-path approach. There is a lack of works in the literature studying the routing as a space of possibilities and seeking in there the optimal point of FEC efficiency.

In this paper we try to present a comparative study across a wide 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. For a given source and destination capillary routing provides numerous multi-path routing suggestions, called *capillary routing layers*. The first layer is the simplest multi-path routing representing max-flow solution. As the layer increases the multi-path routing becomes spreader and more equilibrated. 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. 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. The cooperativeness or friendliness of a routing toward FEC is measured 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] and the authors of [6] and [7] proposed to apply the Adaptive Multi-Rate (AMR) codec, adopted as the standard speech codec by 3GPP, to packet switched networks. Although the proposed end-to-end adaptive FEC mechanism is not aware of the underlying routing and is implemented entirely in the application level of the end nodes, we define the total amount of the adaptive redundancy demanded from the sender during the communication time as a measure of the friendliness and advantageousness of the underlying network routing toward fault-tolerance. We call this measure as Adaptive Redundancy Overall Need (ARON), which is introduced in section III.

Further we measure ARON rating for each layer of capillary routing evaluating the advantageousness of the capillary routing layers toward fault-tolerance. 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 significantly burst the efficiency of FEC. Measurements of the capillary routing friendliness toward fault-tolerance are presented in section IV.

2. Capillary routing

A FEC friendly spread 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. Without requiring any additional capacity a network sustaining several independent single-path streams can re-route all its transmissions to follow multi-path patterns; the FEC redundancy, on the contrary, will require in the multi-path scenario a fewer capacity for recovery the failure of one of the spread paths.

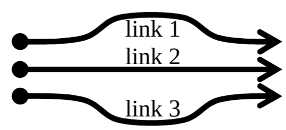

Fig. 1. Transmissions following a single path

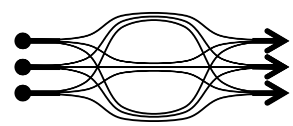

Fig. 2. Transmissions spread over multiple paths do not require extra network capacity

Spread routing alone without FEC would not solve the problem of tolerance. 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. However, most of real-time media streaming applications, oriented to human perception 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, but encoded with a lower rate source coding). VOIP for example can tolerate 8% to 11% packet losses. The constant tolerance can be also obtained or increased by a permanent erasure resilient FEC overhead. For the above examples, a link failure in the single path scenario necessarily drops one of the media streams. At the same time all multiple path streams can in principle sustain a complete failure of any of three links, if are equipped with a strong enough FEC.

It is not however very efficient to permanently transmit with the media stream a large amount of static redundancy. It is much better to increase the redundancy on demand. The receiving node can detect the deficit of delivered packets and demand the sender an appropriate increase of the channel coding. We propose to combine a little permanent constant tolerance of the media stream, combating bursts and weak failures, with an added layer of a dynamic adaptive end-to-end FEC, combating outages, congestions and long failures. The important catalyst of such a static/dynamic FEC for real-time streaming is the diversity of the underlying multi-path routing.

As an approach to the multi-path routing concept we propose a family of capillary routing, which can be best defined by describing the iterative process transforming a simple single-path flow into a capillary route. Already the name of the capillary routing suggests a dispersed communication flow covering numerous paths. In the first layer by minimizing and equilibrating the maximal load of all links, the full mass of the flow is broken across a few parallel routes. Further equilibration is being applied to the remaining portion of the flow by minimizing the maximal load of all links except the bottlenecks of the previous layer. The objective of the second iteration leads to the sub-routes and the sub-bottlenecks of the second layer. Construction and equilibration of the successive layers is continued iteratively 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. At each layer of the capillary routing the flow represents a suggestion of a routing pattern. Each capillary routing layer, growing spreader as the layer number augments, is a subject of evaluation toward the FEC efficiency.

The next figures (from Fig. 3 thru Fig. 6) show three layers of the capillary routing on a network example with 7 nodes.

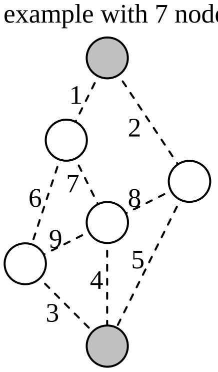

Fig. 3. The top node is the sender and the bottom node is the receiver; all links have a direction from up to down.

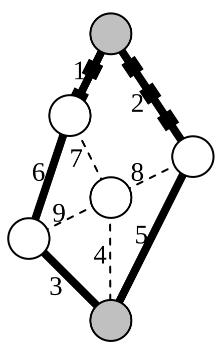

Fig. 4. Links **1**, **2**, 3, 5 and 6 have load of 1/2.

In the first layer of the capillary routing (Fig. 4) we reduce the load of bottleneck links 1 and 2 to their minimal value of $\frac{1}{2}$. Even if links 3, 5 and 6 carry in this solution half of the traffic, they are not bottleneck links and their loads can be further delegated to other links. In the second layer (Fig. 5) we reduce the maximal load of the remaining links to $\frac{1}{3}$, identifying links 3, 4, and 5 as the bottlenecks of the second layer. Link 6 is not a bottleneck link, and its load is further delegated over link 7 in the last third layer (Fig. 6) and is reduced to $\frac{1}{4}$.

In the last solution the load of link 8 is $\frac{1}{6}$ and the load of link 9 is $\frac{1}{12}$. There is no freedom left for the flow after the solution of the third layer, so if we follow the layering, link 8 will be the only bottleneck link of the forth layer and link 9 will be the only bottleneck of the fifth last layer.

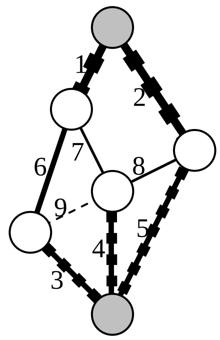

Fig. 5. Links 1 and 2 have load of 1/2; links **3**, **4**, **5** and 6 carry 1/3 of the traffic; links 7 and 8 have load of 1/6.

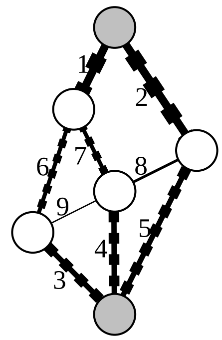

Fig. 6. Links 1 and 2 have load of 1/2; links 3, 4 and 5 have load of 1/3; links **6** and **7** have load of 1/4; link 8 carries 1/6 and link 9 carries 1/12 of the flow.

An LP model for building the capillary routing can be derived straight from the iterative definition of the capillary routing. First minimize the maximal value of the load of links by minimizing an upper bound value applied to all links. Find the bottleneck links of the first layer. Maintaining the first upper bound 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. Find the bottleneck links of the second layer. Minimize the maximal load of remaining links, now also without bottlenecks of the second layer, and continue the iteration.

Although such an 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. The LP problem of the successive layer is obtained by complete removal of bottlenecks from the problem, producing thus new sources and sinks in the network. In the max-flow approach we can also re-calibrate the LP problem before parsing to the next layer avoiding thus the undesirable propagation of precision errors.

In below figures (from Fig. 7 thru Fig. 12), for the same 7-node example we present the capillary routing discovery with the stable LP approach.

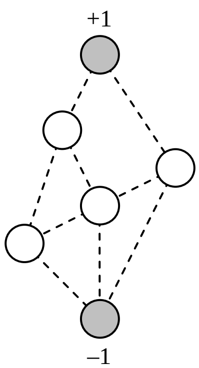

Fig. 7. Initial problem with one source and one sink node

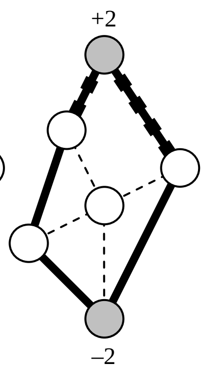

Fig. 8. Maximally increase the flow and adjust the flow out coefficients at nodes, find the bottleneck links

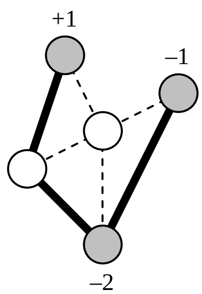

Fig. 9. Remove the bottleneck links from the network adjusting correspondingly the flow out coefficients at the adjacent nodes

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 proportional, through the node's flow out coefficient, to a single variable – the flow increase factor).

Presenting in equations, if 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 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 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.

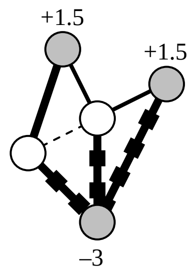

Fig. 10. Maximally increase the flow in the obtained new problem and adjust the flow out coefficients at nodes, find the new bottlenecks

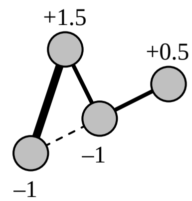

Fig. 11. Remove the bottleneck links from the network adjusting correspondingly the flow out coefficients at adjacent nodes

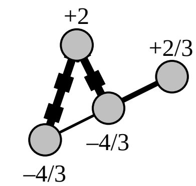

Fig. 12. Again maximize the flow in the obtained new problem adjusting the flow out coefficients at nodes

Only the initial problem (at the first layer) is certainly a unicast problem. All successive layers, in general are synchronous multiple-multicast problems, and thus do not belong to the simple class of "network linear programs" [18].

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 $(i,j) \in B^l$

Description of the bottleneck identification was deliberately left for the end of the section, since behind this single 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 high will be passed to the next iteration. Links undergoing to load reduction under the LP objective are removed from the suspect list and the iteration stops if there are no more links reducing their load.

3. Adaptive Redundancy Overall Need

Adaptive Redundancy Overall Need (ARON), the measure of friendliness of a routing toward fault-tolerance, is defined as the sum of all FEC overheads demanded from the sender for the compensation of sequential losses of all sub-flows carried by each individual link in the footprint of the communication path.

In other words ARON is the routing's proportionality coefficient for the total number of extra redundant packets that the sender will be obliged to transmit during the communication time in order to compensate all arbitrary network failures (assuming a single link failure at a time and a uniform probability and duration of link failures).

In the spread routing, at the node which has multiple outgoing links, the router, according the routing pattern, randomly allocates the packets of the incoming queue to the outgoing links. A temporary congestion or a failure of one of the links of the multi-path routing will produce random losses during the failure time. The packet loss rate observed at the receiver corresponds to the portion of the traffic being still routed toward the faulty route. The streamed real-time media is equipped with a little static tolerance to packet losses (which can be obtained or increased by a proper amount of constant FEC). If the losses measured at the receiver are about to exceed the critical tolerable level the receiver demands the sender via a feed-back channel for an increased rate of extra redundancy. The sender must stream a sufficient quantity of redundancy to compensate the new losses signaled by the receiver maintaining thus the communication at the required level of quality. The sender must compute the redundancy need, the new FEC block size FEC_p, as a function from the percentage of packet losses p reported by the receiver. ARON is computed using values of FEC_p for the loads of all links in the communication path. 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.

One parameter of streaming is M, the number of media packets in the block, another parameter we must rely on for computing the transmission block size $FEC_p \geq M$ is the Decoding Error Rate (DER) – the desired decoding failure probability at the receiver.

If we have 20 media packets in each transmission block and 20% random losses in the network, the mean number of received packets is 16. If we add 5 redundant packets and transmit for each 20 media packets a block of 25 packets the mean number of successfully received packets will be 20, which is the number needed for the recovery of the original media. However the probability of receiving 19 packets or 18 packets (which makes the decoding impossible) is quite high. Therefore for small values of M, which is the limiting factor of the two-way 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. Thus the mean of received packets should be much higher than M and the probability of receiving less than M packets must be maintained very low, i.e. $FEC_p \geq M/(1-p)$.

The probability of having 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:

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

Thus the above formula computes 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 function from the random loss probability $0 \leq p < 1$.

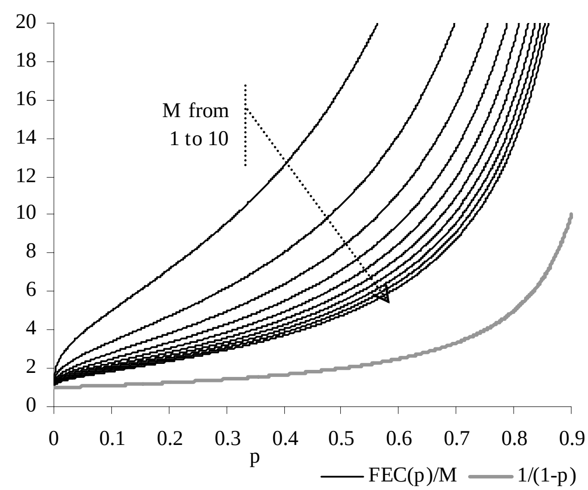

Fig. 13. FEC overhead as a function from the packet loss rate

Several FEC_p / M functions (FEC overheads) for media packets from 1 to 10 are plotted in the chart of Fig. 13 (for $DER = 10^{-5}$). These functions are compared with $\frac{1}{1-p}$, 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. Thus, if necessary the playback buffer must be extended enough to consider also the packet holding resulting from the decoding needs. 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.

Let L be the set of links laying on the communication path, and let $r(l)$ be the load of a link $l \in L$ under the present routing scheme. Let the percentage $0 \le t < 1$ of recoverable losses, be the static level of the tolerance permanently maintained in the media stream (even if there are no losses). The equation for ARON then looks as follows:

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

Under Shannon capacity assumption:

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

Shannon capacity can be assumed with capacity approaching fountain codes [19] and very long buffers. An example for which the ARON equation with Shannon capacity assumption is applicable is the problem of a last kilometer bottleneck for the internet or wireless MBMS user downloading and watching a one way video from multiple servers (studied by [14] and [20]). Generally, ARON equation assuming Shannon capacity must be used for file transfer (or very large block) applications in which the reception must be maintained at a constant rate (the bandwidth of the last kilometer) and the file transfer must be accomplished within the time of the file size over the reception rate. Thus 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 a real-time application, however, the first equation must be used with an appropriate choice of DER and the number of media packets M carried in the FEC block.

A simple scalar value of ARON can rate the friendliness of a network routing toward the FEC efforts of a streaming application. Within a number of suggested network routings, the best is the one, whose ARON is the smallest.

4. Measuring the friendliness of capillary routing

The efficiency of the capillary routing approach will be weighted when we rate the routings suggested by each capillary layer. The first reference layer of the capillary routing, having the footprint of the max-flow solution, is a routing with an equal load balance of the main mass of the flow.

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 capillary routing schemes that are subject of comparisons 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, but we rather study the advantageous trends within the scope of multi-path routing approaches.

We need to compute average ARON of numerous network samples to evaluate the overall performance of the capillary approach, rather than to measure a particular routing example. 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. 14 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.

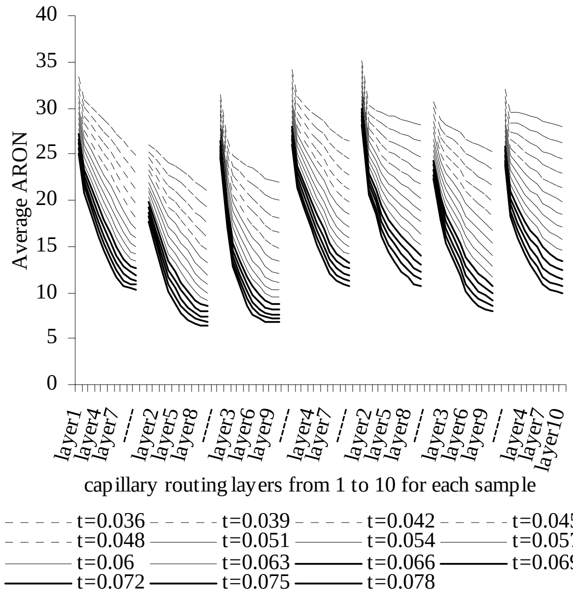

Fig. 14. Average ARON as a function from the capillary routing layer

The permanent tolerance of any application (even with no initial tolerance like in a file transfer) can be obtained by adding a proper amount of constant FEC. 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, only the first few layers 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}.

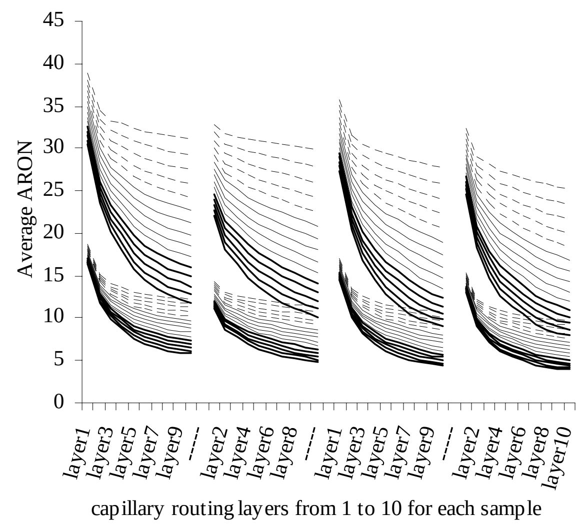

Fig. 15. Average ARON computed assuming real-time streaming (the group of curves above) and Shannon capacity (the group below)

In the example of Fig. 15 the network consists of 120 nodes and there are 150 timeframes of another random-walk Ad-Hoc network. The corresponding 150 network samples are broken into four sets. The group of 15 curves below corresponds to ARON ratings computed assuming Shannon capacity and the group of 15 curves above corresponds to ARON assuming an MDS code with the media streaming parameters identical to those of the previous example ($M = 20$, $DER = 10^{-5}$). The range of static tolerances of media streams is the same, from 3.6% to 7.8% of tolerable packet losses. As expected ARON computed with the Shannon capacity assumption is much lighter (requiring less FEC effort from the sender) compared with ARON of real-time media with limited buffer; but both of them follow the trend of a noticeable improvement as the underlying routing grows spreader.

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 even 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.

In our approach, the routing is independent from the media application; it is just that the routing can be friendly or hostile toward the fault-tolerance. Guided by the capillary routing patterns ISP, wishing to be media streaming friendly, can upgrade the routing of its network to be as spread as possible, focusing also on the seemingly non important non-bottleneck portions of its network. 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.

5. 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.

6. 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