Fault-Tolerance of Capillary Multi-Path Routing for Real-Time Multimedia Streaming with Forward Error Correction

Emin Gabrielyan
EPFL and Switzernet Sàrl
Lausanne, Switzerland
emin.gabrielyan@switzernet.com

Roger D. Hersch
Swiss Federal Institute of Technology – EPFL
Lausanne, Switzerland
rd.hersch@epfl.ch

Abstract

Using forward error correction (FEC) in off-line streaming, permitting large buffers, yields spectacular results, but real-time streaming puts hard restrictions on the buffer size and therefore does not allow FEC to deal with long link failures on a single path route. Multi-path routing however can make FEC efficient also for real-time streaming. In this paper we introduce a 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 can be measured by Adaptive Redundancy Overall Need (ARON), which is proportional to the sender's total channel coding effort needed for recovering the failure of each individual link in the multi-path route. A dozen of capillary routing layers are built on several hundreds of network samples obtained from a random walk wireless Mobile Ad-Hoc Network (MANET). Rating of these routing suggestions with ARON shows that the FEC friendliness improves substantially as the spreading of the routing grows.

1. 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 forward error correction FEC yields spectacular results in the most challenging network conditions. 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 codes [2] enable a fast delivery of the day's film footage over the lossy internet across thousands of miles. Third Generation Partnership Project (3GPP), recently adopted Raptor as a mandatory code in Multimedia Broadcast/Multicast Service (MBMS), for its significant performance in file transfer and linear CPU cost [3], [4], [5].

The above examples of off-line streaming can significantly benefit from FEC due to the fact that in contrary to real-time streaming, the application is not obliged to deliver in time the "fresh" packets 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 applications of FEC to real-time streaming. According to [6] improvements due to the application of FEC are present only if the stochastic packet losses range is between 1% and 5%. Author of [7] proposed to combine FEC with retransmissions to improve its efficiency in real-time packetized streaming. In [8], a high overhead has been reported 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 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 efficient for fault-tolerant real-time streaming. 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. The routing topologies, so far, were not regarded as a space of solutions and a ground for searching a FEC effective pattern.

In this paper we try to present a comparative study of FEC friendliness on various multi-path routing patterns. Single path routing, being too hostile, is excluded from our comparisons.

As means for achieving multi-path routing, we propose *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, spreading across more links as the number of layer increases. The first layer is a simple multi-path routing representing a max-flow solution. Successive layers are obtained by recursively spreading the individual sub-flows of previous layers. With the capillarization of the routing, the communication uses the network more reliably employing more of alternative transmission capacities offered by the network topology. The last layer represents the complete capillary routing

and, in contrast to shortest path or max-flow methods, has one unique solution for any given network. We present the capillary routing construction in section 2.

To compare two multi-path routing suggestions, we are introducing a measure 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 at the application level at the end nodes. By default, the sender is streaming the media 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 increases the FEC overhead whenever the packet loss rate is about to exceed the tolerable limit. For measuring the friendliness of a specific routing pattern we use Adaptive Redundancy Overall Need (ARON), which represents the total amount of the failure recovery adaptive redundancy required from the sender during the communication time. The novelty brought by ARON, introduced in details in section 3, is that a routing topology of any complexity can be rated by a single scalar value.

In section 4, 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 for multi-path routing, significant improvement can be obtained by further improving the basic path diversity provided by the first layer of the capillary routing (i.e. by the max-flow solution). Multi-path routing, similarly to buffering, can substantially burst the efficiency of FEC.

2. Capillary routing

Capillary routing seeks to minimize the impact of individual link failures, giving thus the encoder a greater chance of recovering the failure.

The strategy for capillary routing can be best defined by describing an iterative Linear Programming (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. Minimize the maximal load of the 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 (Fig. 1 to Fig. 3). The top node on the diagrams is the sender and the bottom node is the receiver; all links are oriented from top to bottom.
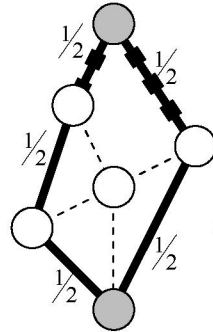

Fig. 1. In the first layer the flow is equally split over two paths, which has two bottlenecks, marked by thick dashes.
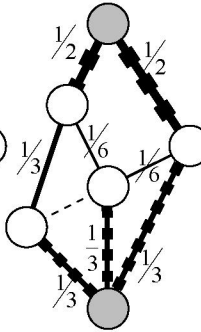
Fig. 2. The second layer minimizes the maximal load of the remaining seven links and finds three bottlenecks, each with a load of 1/3.
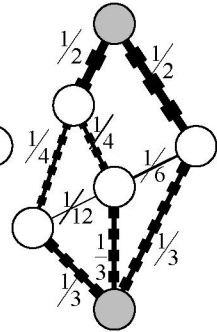
Fig. 3. The third layer minimizes the maximal load of the remaining four links and finds two bottlenecks each with a load of 1/4.

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 $\frac{1}{4}$. The two remaining links have their loads ultimately fixed to $\frac{1}{6}$ and $\frac{1}{12}$.

Although the LP approach derived from the definition of the capillary routing is fully valid, precision errors propagate through the sequence of LP problems often reaching noticeable sizes and resulting in infeasible LP problems. We have found a different, stable LP method maintaining the parameters and variables always in a similar range.

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

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 a set of sources to a set of sinks, where all rates of transmissions by sources and all rates

of receptions by sinks increase proportionally (i.e. synchronously) in respect to each node's flow-out coefficient (either positive or negative).
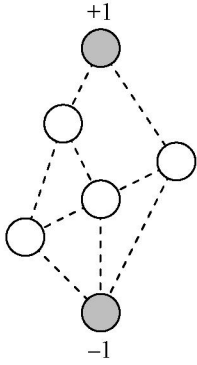

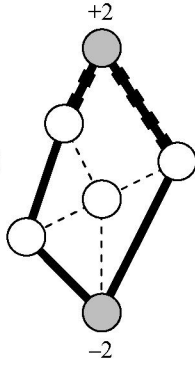**Fig. 4.** Initial problem with one source and one sink node

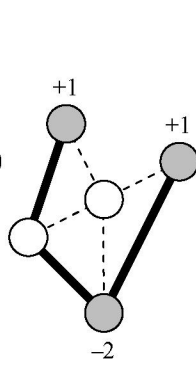**Fig. 5.** Maximize the flow, fix the new flow-out coefficients at the nodes and find the bottleneck links

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

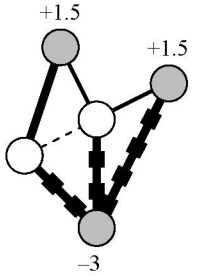

Fig. 7. Maximize the flow in the new sub-problem, fix the new flow-out coefficients at the nodes and find the new bottlenecks
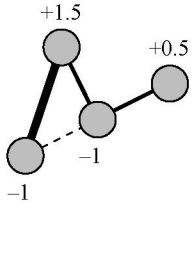
Fig. 8. Remove the bottleneck links from the network and adjust correspondingly the flow-out coefficients at the adjacent nodes
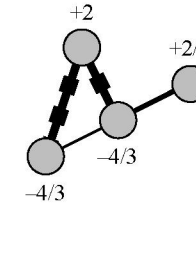
Fig. 9. Again maximize the flow in the obtained new problem and fix the new resulting flow-out coefficients at the nodes

The LP problem at each successive layer is obtained by the complete removal of the bottlenecks from the previous LP 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]. Let us present the iteration as equations:

We define the flow problem of a synchronous multiple-multicast at layer l 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 $i \in N^l$

The max-flow solution of the problem yields the flow increase factor F^l and the set of bottlenecks B^l, where $B^l \subset L^l$. In the above example (**Fig. 4** to **Fig. 9**) the flow increase factors are $F^1 = 2$, $F^2 = 1.5$ and $F^3 = 4/3$.

Then for the next successive layer: N^{l+1}, L^{l+1} and f^{l+1} are defined as follows:

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

$$\underset{\substack{\text{add 1 for each incoming} \\ \text{bottleneck link } (i, j)}}{} \quad \underset{\substack{\text{subtract 1 for each} \\ \text{outgoing bottleneck } (j, k)}}{}$$

After a certain number of applications of the max-flow optimizations, we will ultimately 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 are enclosed in bottlenecks of one of the layers.

To restore the proportions we must now divide the flow values of the max-flow solutions by all preceding flow increase factors. The true value of flow $r_{i,j}$ traversing the bottleneck link $(i, j) \in B^l$ of layer l is the initial one unit of flow divided by the product of the flow increase factors of layer l and all previous layers:

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

The max-flow approach proves to be very stable, because it maintains all values of variables and parameters within a close range of unity (even for very deep layers with tiny loads) and also because it enables to re-calibrate the LP problem (taking into account the flow conservation rules) before the next pass, avoiding thus the undesirable propagation of precision errors.

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 at the initial maximal level will be passed to the next iteration. Links whose load is reduced under the LP objective are removed from the suspect list and the bottleneck hunting iteration stops if there are no more links to remove.

3. Adaptive Redundancy Overall Need (ARON)

Without FEC spread routing alone would not solve the problem of tolerance. Most 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 also be obtained or increased by a constant FEC code. We propose to combine the little static tolerance of the media stream, enabling recovering from weak failures, with a dynamically added adaptive FEC, combating the strong failures exceeding the tolerable packet loss rate.

ARON is the sum (over all links) of percentages by which the sender increases its transmission rate to

combat every individual link failure (in the communication footprint). 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 $5 \cdot 25\% = 1.25$.

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

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

Let us consider 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 individual network failures occurring during the total 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 schemes between the end nodes; ARON in contrast is the component depending on the size and the topology of the particular multi-path routing. In other words, in respect to the total number of extra redundant packets or in respect to the cost of the redundancy in communication, ARON is the proportionality coefficient representing solely and fully the routing scheme.

Redundant packets are injected into the original real-time stream for every block of M media packets using systematic erasure resilient codes (thus without affecting the original media packets). During the streaming, the number of media packets (M), limited by the receiver's playback buffer time, is supposed to stay constant. The number of redundant packets for each block of M media packets is however variable. 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 order to response to a packet loss rate p. We are assuming that the media stream has a static tolerance to losses $0 \le t < 1$ by maintaining a constant FEC code, which by default streams the packets in FEC blocks of length of FEC_t. When the loss rate p measured at the receiver is about to exceed the tolerable limit t, the sender must increases 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, 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 relative traffic load of $0 \le r(l) \le 1$ will produce at the receiver a random packet loss rate equal to the same relative traffic load $r(l)$. The equation (1) can thus be re-written as follows:

$$ARON = \sum_{l \in L \, | \, t \le r(l) < 1} \frac{FEC_{r(l)} - FEC_t}{FEC_t} \quad (2)$$

According to the sum index, the critical links of the path carrying the entire traffic are ignored. The FEC required for the compensation of long failures of such links would be infinite. Since any routing suggestion will unavoidably pass its entire traffic through the critical link, removing the corresponding infinite component from ARON cannot therefore affect our comparisons. From the other side, a non-critical link will never carry the entire traffic in all considered here routing suggestions.

We compute the FEC_p function assuming a Maximum Distance Separable (MDS) code, e.g. Reed-Solomon code. By the choice of an MDS code, the reception of the sufficient number 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.

In order to compute the transmission block size $FEC_p \ge M$, we must fix the desired Decoding Error Rate (DER), i.e. the acceptable decoding failure probability at the receiver.

To collect mean M packets at the receiver we must transmit $\frac{M}{1-p}$ 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 we must send much more redundant packets in the block than is necessary to receive mean M packets at the receiver side. The average 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 according to the binomial distribution:

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

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

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

This 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.

Several FEC_p / M functions (FEC overheads) for media packets from 1 to 10 are plotted in Fig. 10 (for $DER = 10^{-5}$). These functions are compared with a theoretical rate $\frac{1}{1-p}$, assuming the Shannon capacity conditions for erasure channels. Higher is the number of media packets in the block (i.e. longer is the buffering time) closer the communication can approach to theoretical limit.
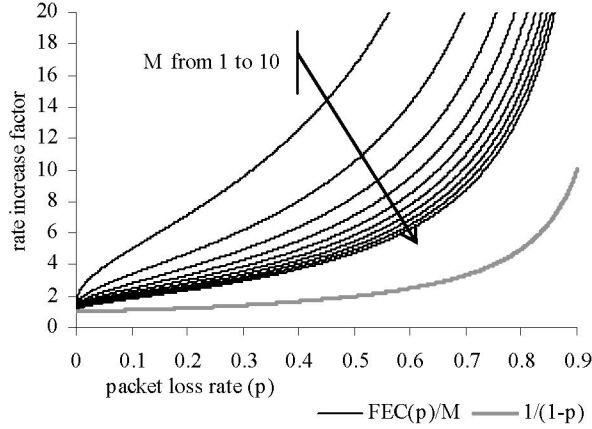
Fig. 10. FEC overhead as a function from the packet loss rate ($DER = 10^{-5}$)

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 large enough to hold also packets of the FEC block (at least M packets for MDS codes). 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 transfers. An example of nearly off-line application requiring path diversity is the problem of the efficient usage of the last kilometer bottleneck for an Internet user downloading a file or watching a one-way video. Streaming from multiple servers was studied in [14] and [20]. The buffering time can be a few minutes long, with thousands of media packets within a single FEC block. Capacity approaching fountain codes [19] are the best for this kind of application. Path diversity is needed, because during internet failures the reception can be maintained at the receiver thanks to the streaming also through alternative paths. To ensure the reception at the highest rate of the last kilometer bottleneck, during internet failures the sender must dynamically increase its rate.

For this particular application, assuming the near optimal Shannon with $FEC_t = M/(1-t)$ and $FEC_{r(l)} = M/(1-r(l))$, we derive from equation (2) the following relationship for ARON:

$$ARON = \sum_{l \in L \,|\, t \leq r(l) < 1} \left(\frac{1-t}{1-r(l)} - 1 \right) \qquad (3)$$

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

4. Measuring the friendliness of capillary routing

In order to evaluate the overall performance of the capillary approach we compute the average ARON ratings on hundreds of network samples. First, we consider the first layer routing scheme for each network sample and obtain thus the average ARON rating of the first layer routing (i.e. of the max-flow suggestion). Then we compute the second layer routing individually for each considered network sample in the same set and we obtain the set's average ARON rating for the routing suggestions of the second layer. We measure 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 we have seven sets, each containing 42 network samples. At the same time we consider also 15 media streams different by their constantly maintained tolerance to losses, varying from 3.6% to 7.8%. Thus for each set we have 15 curves of average ARON ratings. All of them decrease as the capillary routing layer increases from 1 thru 10 demonstrating the improvements delivered by the higher layers. Although the spreading of the flow and the further recursive sub-spreading of the sub-flows increase the number of links and therefore also the rate of failures in the communication footprint, the ARON cost however sensibly reduces from such strategy and therefore does also the overall FEC effort of the sender.
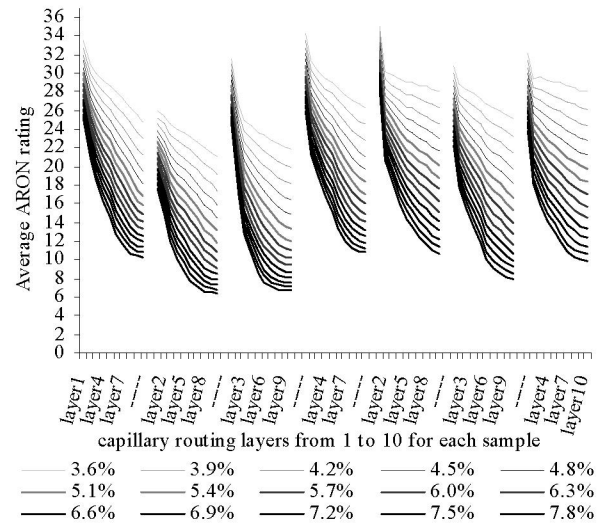

Fig. 11. Average ARON as a function of the capillary routing layer

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 each additional unit of a little static tolerance in the media stream stresses stronger the efficiency gain achieved by the deeper layers. 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 of the layer, depends on the distance between the end nodes, 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 a 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 of each other) then there is a link between them. In the above example, there are 115 nodes and 300 timeframes each producing a different network sample (all of which are broken into 7 sets presented 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 accessed, a capillarization of the 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] and [21]).

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

5. Conclusion

We introduce multi-path capillary routing, which is built layer by layer. The first layer provides a simple max-flow solution, but as the layer number increases the underlying routing, spreads out and uses the network more securely. We also introduced ARON for measuring the friendliness of a multi-path routing suggestion by a single scalar value. The ARON rating corresponds to the total encoding effort that the sending node needs to provide for combating the losses occurring from the non-simultaneous failures of links in the communication path. By rating the layers of the capillary routing on hundreds of network samples, we have shown that by recursively diversifying the flow and the sub-flows of the routing, we can substantially improve the FEC friendliness of multi-path routing.

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

[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