2006 6th International Conference on ITS Telecommunications Proceedings

Rating of Routing by Redundancy Overall Need

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

Roger D. Hersch
École Polytechnique Fédérale de Lausanne - EPFL
Switzerland
rd.hersch@epfl.ch

Abstract – **Thanks to the large buffering time of off-line streaming applications, erasure resilient Forward Error Correction (FEC) codes can improve the reliability of communication particularly well. However real-time streaming puts hard restrictions on the buffer size making FEC inefficient for combating long link failures on single path routes. Path diversity is orthogonal to buffering and permits real-time streaming to also benefit from application of FEC. We introduce a capillary routing algorithm offering layer by layer a wide range of multi-path routing topologies starting from a simple solution and evolving toward reliable routing patterns with highly developed path diversity. The friendliness of a multi-path routing pattern is rated by the overall amount of FEC redundancy required for combating the non-simultaneous failures of all links of the communication footprint. We rated the friendliness of a dozen of capillary routing layers, built on several hundreds of network samples obtained from a random walk Mobile Ad-Hoc Network (MANET). The overall requirement in redundant FEC codes decreases substantially as the spreading of the routing grows.**

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 can mitigate packet losses.

In off-line packetized applications Forward Error Correction (FEC) dramatically improves the quality and performance of communications under challenging network conditions [1]. Thanks to erasure resilient Raptor codes [2] it is possible to simultaneously update voluminous GPS maps of millions of motor vehicles without a feedback via a satellite broadcast channel under conditions of arbitrary fragmental visibility. In the film industry, LT codes [3] enable a 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, much faster than with FedEx or DHL. These examples of off-line streaming significantly benefit from FEC due to the fact that contrary to real-time streaming, the application is not obliged to deliver in time to the user the "fresh" packets and the buffer size is not a concern. When the real-time streaming constrains restrict the buffer size, FEC can only mitigate short granular failures. Studies reporting weak or negligible improvements when applying FEC to real-time streaming [4], [5], [6] and [7] assume that the media stream follows a single path.

By exploiting path diversity, "replacing" the time diversity offered by long buffering, FEC can efficiently improve the fault-tolerance of also real-time streaming. There is an emerging body of a literature showing the applicability of FEC in real-time streaming [8], [9], [10] and [11] when an alternative path is available. However these studies address the comparison of the single path solution with the path diversity approach and the routing patterns are limited to either two (possibly correlated) paths or in the best case to a sequence of parallel and serial links.

We present a comparative study over a wide range of multi-path routing patterns. Single path routing, being considered as too hostile, is excluded from our comparisons. Steadily diversifying multi-path routing patterns are created with *capillary routing* algorithm, offering layer by layer various routing suggestions (sections III).

For evaluating a multi-path routing suggestion, we rely on the overall amount of adjustable FEC codes needed for combating non-simultaneous failures of all individual links in the communication path. Adjustable FEC for real-time streaming was proposed by several authors [4], [5] and [6]. In two-way real-time streaming, the packet loss rate information is usually transmitted with the opposite flow using Real-time Transport Control Protocol (RTCP). The sender increases the FEC overhead whenever the packet loss rate is about to exceed the tolerable limit. The friendliness of the underlying network routing is measured by Failure Recovery Redundancy Overall Requirement (FRROR) introduced in section II. The novelty brought by FRROR is that a routing topology of any complexity can be rated by a single scalar value.

By rating each layer of capillary routing with FRROR, in section IV we evaluate the decrease in the overall requirement of FEC codes achieved by capillarization of the routing path. Network samples are obtained from a wireless random walk Mobile Ad-hoc Network (MANET) with several hundreds of nodes. We show that capillarization, similarly to buffering, can substantially burst the effective value of FEC.

II. REDUNDANCY OVERALL REQUIREMENT

Most real-time media streaming applications are tolerant to a certain level of packet losses due to passive error concealment or media encoding techniques. Voice over IP (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 a little static tolerance for combating weak failures, with a dynamically added adaptive FEC combating the strong failures exceeding the tolerable packet loss rate.

For a given routing pattern FRROR is defined as the sum of all FEC transmission rate increase overheads required from the sender to combat all individual link failures. For example, if the communication footprint consists of five links, and in

0-7803-9586-7/06/$20.00 ©2006 IEEE.

response to each individual link failure the sender increases the packet transmission rate by 25%, then FRROR will be equal to the sum of these five FEC transmission rate increases, i.e $FRROR = 5 \cdot 25\% = 1.25$.

Redundant packets (of almost the same size as the source media packets) are injected in the original media stream for every chunk of *M* source packets using a systematic erasure resilient code (thus without affecting the source media packets). During streaming the number of media packets (*M*) in each chunk is supposed to stay constant. The number of redundant packets for each chunk of *M* media packets is however variable, depending on the conditions of the erasure channel. The *M* media packets with their related redundant packets form a FEC block. By $FEC_p \geq M$ we denote the FEC block size chosen by the sender in response to a packet loss rate *p*. We are assuming that the media stream has also a static tolerance to losses $0 \leq t < 1$ obtained with a constant FEC code, meaning that by default the packets are streamed 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 increases its transmission rate by injecting additional redundant packets.

The random packet loss rate, observed at the receiver during the time of a complete failure (or a full congestion) of a link, is the portion of the traffic being routed toward the faulty link. Thus a complete failure of a link *l*, carrying according to the given routing pattern a relative traffic load of $0 \leq r(l) \leq 1$, will produce at the receiver a random packet loss rate equal to the same relative traffic load $r(l)$. The equation for FRROR can thus be written as follows:

$$FRROR = \sum_{l \in L \,|\, t \leq r(l) < 1} \left(\frac{FEC_{r(l)}}{FEC_t} - 1 \right) \tag{1}$$

The links carrying the entire traffic are skipped in the sum index of equation (1), since the FEC required for the compensation of failures of such links is infinite. If for a given network topology the link is critical, any routing suggestion will unavoidably pass its entire traffic through that link, and therefore without affecting the comparison, the corresponding "equivalent" infinite components can be removed from the FRROR rates of all suggested routings. By construction (section III), none of considered multi-path routing schemes passes its entire traffic through a non-critical single link.

The FEC_p function we compute assuming a Maximum Distance Separable (MDS) code (e.g. a Reed-Solomon code). By the choice of an MDS code, the condition for a successful decoding of all original source packets carried by a FEC block is the reception of exactly the same number (*M*) of packets (of any type: media or redundant) as there were original media packets in the block.

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

According to the binomial distribution, equation (2) gives the decoding failure probability δ at the packet loss rate *p* if the FEC block size is equal to *N*.

$$\delta = \sum_{n=N-M+1}^{N} \binom{N}{n} \cdot p^n \cdot (1-p)^{N-n} \tag{2}$$

For computing the carrier block's minimal length for a satisfactory communication, it is sufficient therefore to steadily increase in equation (2) the carrier block length *N* until the desired decoding error rate (DER) is met.

The receiving side of the media application is equipped with a playback buffer in order to compensate for the network jitter and to reorder packets arriving in the wrong order. The receiver must also hold in the playback buffer enough packets to restore the recoverable losses. The larger the number of media packets *M* in the FEC block, the smaller the cost of FEC overhead is. For example in VOIP with a 20 ms sampling rate (g729r8 or AMR codec) the number of media packets *M* in a single FEC block must not exceed 20 – 25 packets (each carrying one sample).

III. CAPILLARY ROUTING

Capillary routing seeks to minimize the impact of individual link failures on real-time streaming, thus requiring less effort from the encoder for recovering the failure.

The capillary routing algorithm is defined by 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 all links by minimizing an upper bound value applied to all links. In the first layer, the full mass of the flow is split equally across the main parallel routes. Find the bottleneck links of the first layer. By maintaining the first upper bound (applied to all links) on its minimal level, minimize the maximal load of the remaining links by minimizing a new upper bound value applied to all links except the bottleneck links of the first layer. The second iteration discovers the sub-routes and the sub-bottlenecks of the second layer. Then, minimize the maximal load of the remaining links, now also without the bottlenecks of the second layer (maintaining the first and the second upper bounds at their lowest levels), 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.

Although the described LP process is completely valid, it is numerically instable, the precision errors, propagating through the layers of capillary routing, reach noticeable sizes and, when dealing with tiny loads, result in infeasible LP problems. We have found a different, stable LP method maintaining the values of parameters and variables always in the same scale.

Instead of decreasing the maximal value of loads of the links, the routing path is discovered by solving max flow problems. The resulting routing solutions of these two methods are identical except that the proportions of flow differ by the increase factor of the max flow solution. The diagrams of Fig. 1 to Fig. 3 show for a toy case the discovery of the first three layers of capillary routing according to the max-flow LP approach.

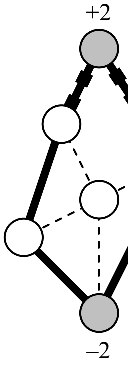

Fig. 1. Maximize the flow of the initial problem with one source and one sink, fix the new flow-out coefficients at the nodes and find the bottleneck links (layer 1, $F^1 = 2$)

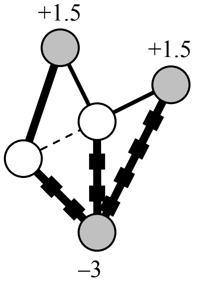

Fig. 2. Remove the bottleneck links from the network, adjusting the flow-out coefficients at the adjacent nodes, maximize the flow in the new sub-problem, fix the new flow-out coefficients at the nodes and find the new bottlenecks (layer 2, $F^2 = 1.5$)

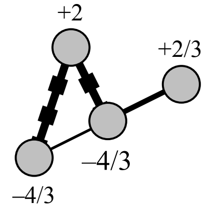

Fig. 3. Again remove the bottleneck links of the previous solution from the network, again maximize the flow in the obtained new problem, fixing the new flow-out coefficients and find the new bottlenecks (layer 3, $F^3 = 4/3$)

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 (Fig. 1). At each subsequent layer we have a bounded multiple-sources/multiple-sinks 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 in respect to each node's flow-out coefficient (either positive or negative). The LP problem at each successive layer is obtained by 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. Except for the unicast problem of the first layer, the successive layer problems do not belong in general to the simple class of "network linear programs" [12].

If, we define the bounded multiple-sources/multiple-sinks problem at layer l by the sets of nodes and links and by the flow-out parameters for sources and sinks (all indexed with an upper index l) as follows:

- set of nodes N^l,
- set of links $(i,j) \in L^l$, where $i \in N^l$ and $j \in N^l$,
- flow-out values f_i^l for all $i \in N^l$
- at layer l the max-flow solution yields the flow increase factor F^l and the set of bottlenecks B^l, where $B^l \subset L^l$

Then, the equations for computing the sets N^{l+1}, L^{l+1} and the flow out parameters f^{l+1} of the next layer are as follows:

- $N^{l+1} = N^l$
- $L^{l+1} = L^l - B^l$
- $f_j^{l+1} = f_j^l \cdot F^l \quad + \underset{\text{add 1 for each incoming bottleneck link } (i,j)}{\sum_{(i,j)\in B^l}(+1)} \quad + \underset{\text{subtract 1 for each outgoing bottleneck } (j,k)}{\sum_{(j,k)\in B^l}(-1)}$

After a certain number of applications of the max-flow objective with corresponding modifications of the problem, we 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 are enclosed in bottlenecks of one of the layers.

To restore the original proportions of the flow, the flow increases by the preceding max-flow solutions must be all compensated. The true value of flow $r_{i,j}$ traversing the bottleneck link $(i,j) \in B^l$ of layer l is the initial single unit of flow divided by the product of the flow increase factor F^l of layer l with the flow increase factors F^i (where $1 \le i < l$) of all preceding 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 validate and if necessary re-calibrate the flow-out parameters of the LP problem formulated for the next layer of capillary routing.

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 the traffic over all links having maximal load and being suspected as bottlenecks. Only links maintaining their load at the initial maximal level will be passed to the next iteration of the hunting loop. Links whose load has been reduced under the LP objective are not bottlenecks and removed from the list of candidates. The bottleneck hunting loop stops if there are no more links to remove.

IV. FRIENDLINESS OF CAPILLARY ROUTING

We compute the average FRROR rating for various network samples in order to evaluate the overall performance of the capillary approach. First, we consider the first layer routing scheme for each considered network sample and obtain thus the average FRROR rating for all routing (max-flow) schemes of the first layer. Then we compute the second layer routing individually for each considered network sample and obtain the average FRROR rating for the routing suggestions of the second layer. We measure the average FRROR rating for the capillary routing layers 1 to 10 and show its decrease as the layer number grows.

In Fig. 4, we have eight sets of network samples, each containing 25 network samples. At the same time we consider also 8 media streams which differ by their static tolerance to losses varying from 3.3% to 7.5%. Thus for each set we have 8 curves of average FRROR ratings. All of them decrease as the capillary routing layer increases from layer 1 to 10 demonstrating the improvements due to the stronger capillarization.

Although spreading out of the flow uses more links and therefore also increases the total rate of failures in the communication footprint, capillarization of the multi-path routing reduces however the total FRROR rating considerably and therefore also the overall FEC effort of the sender combating the link failures and packet losses.

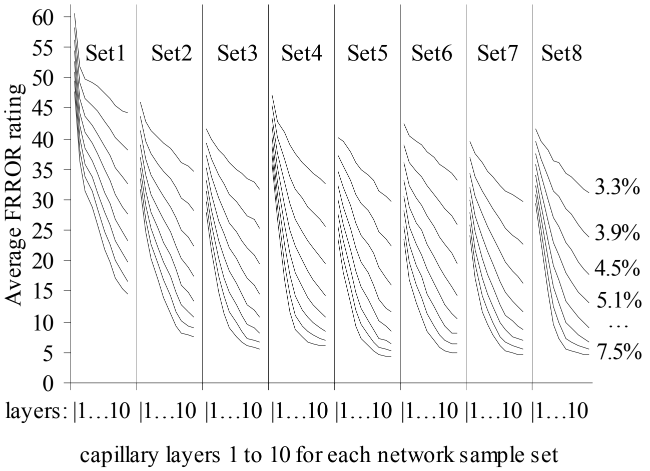

Fig. 4. Average FRROR as a function from the capillary routing layer (the static tolerance of the stream from 3.3%, for the upper curve, to 7.5%, for the lower curve, by a step of 0.6%)

Logically, the FRROR curve of the media stream is shifted down as the statically added tolerance increases. At the same time it is interesting to observe that, in contrast to a weak static tolerance, the presence of a higher static tolerance yields a much stronger efficiency gain achieved by the deeper routing layers.

The pattern of the FRROR curve, as a function of the layer, depends on the distance between the peers, the network size and its density. The network samples for the above chart are drawn 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) then there is a link between them. In the above example, there are 300 nodes and 200 time-frames, each leading to a separate network sample (all of which are distributed into eight sets represented on the above chart).

The FRROR rating of routing samples is computed by equation (1), where the FEC block size (as function of packet loss rate p) is computed based on equation (2). The number of media packets (M) per transmission block is 20 and the desired decoding failure rate (DER) is 10^{-5}.

V. CONCLUSIONS

The quality and reliability issues concerning real-time streaming over packet networks are of growing importance. Commercial real-time streaming applications however do not consider channel coding as a serious solution for improving the reliability of communication since in single path transmission, even heavy FEC overheads cannot protect against failures lasting more than the short duration of the playback buffer. Recent studies demonstrated that path diversity makes FEC applicable for real-time streaming. By studying a wide range of routing topologies, we show that the proper choice of the routing pattern can make FEC extremely efficient. Combination of channel coding with appropriate multi-path routing improves the reliability of real-time packetized communications even in the case of very short playback buffers.

We introduce multi-path capillary routing, built layer by layer. The first layer provides a simple max-flow solution, but as the layer number increases the spreading of the underlying routing scheme makes the network more secure for real-time media streaming. We introduce FRROR, a method for rating multi-path routing patterns by a single scalar value. The FRROR rating corresponds to the total redundancy overhead that the sending node provides in order to combat the losses occurring from non-simultaneous failures of links in the communication path. We show a substantial improvement of the routing topology and decrease of the required amount of FEC codes, as the capillarization of the routing developes.

Capillary fault-tolerant routing can be applicable to Ad-Hoc or sensor networks, 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, path diversity can be still obtained relying on overlay networks using peer-to-peer relay nodes [11] and [13].

VI. REFERENCES

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

[2] Amin Shokrollahi, “Raptor codes”, ISIT’04, Jun 27 - Jul 2, p. 36

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

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

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

[6] 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

[7] 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

[8] 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”, GLOBECOM’04, Vol. 2, Nov 29 - Dec 3, pp. 977-981

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

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

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

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

[13] 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