

Adaptive Causal Network Coding with Feedback for Multipath Multi-hop Communications

Alejandro Cohen*, Guillaume P Thiran[†], Vered Bar Bracha[‡], Muriel Médard*,

*Research Laboratory of Electronics, MIT, Cambridge, MA, USA, {cohenale, medard}@mit.edu

[†]UCLouvain, Belgium, guillaume.thiran@student.uclouvain.be

[‡]Intel Corporation, vered.bar.bracha@intel.com

Abstract—We propose a novel multipath multi-hop adaptive and causal random linear network coding (AC-RLNC) algorithm with forward error correction. This algorithm generalizes our joint optimization coding solution for point-to-point communication with delayed feedback. AC-RLNC is adaptive to the estimated channel condition, and is causal, as the coding adjusts the retransmission rates using a priori and posteriori algorithms. In the multipath network, to achieve the desired throughput and delay, we propose to incorporate an adaptive packet allocation algorithm for retransmission, across the available resources of the paths. This approach is based on a discrete water filling algorithm, i.e., bit-filling, but, with two *desired objectives*, maximize throughput and minimize the delay. In the multipath multi-hop setting, we propose a new decentralized balancing optimization algorithm. This balancing algorithm minimizes the throughput degradation, caused by the variations in the channel quality of the paths at each hop. Furthermore, to increase the efficiency, in terms of the desired objectives, we propose a new selective recoding method at the intermediate nodes. Through simulations, we demonstrate that the performance of our adaptive and causal approach, compared to selective repeat (SR)-ARQ protocol, is capable of gains up to a factor two in throughput and a factor of more than three in delay.

I. INTRODUCTION

The increasing demand for network connectivity and high data rates necessitates efficient utilization of all possible resources. In recent years, the connectivity moved forward from point-to-point schemes (i.e. single path, SP) to heterogeneous multipath (MP) multi-hop (MH) networks in which intermediate nodes can cooperate and share the common medium for efficient communications. Such advanced networks tend to provide robustness and reliable communications with high data rates by simultaneously sharing the physical layer resources over each path and hop. However, it is critical to ensure that efficiency is not compromised and that in-order delivery delay is managed [1], particularly when the links are unreliable and retransmissions are required; when there are high round-trip-time (RTT) delays between the sender and the receiver; or when the state of each link/path is not fully determined. Figure 1 illustrates the considered settings of communications.

To achieve the min-cut max-flow capacity in the multipath multi-hop networks, solutions based on information-theory with a very large blocklength regime have been considered [2]. However, in streaming communications, which are utilized, for example, in audio/video transmissions, automotive, smart-city, control applications, etc., there are strict real-time constraints that demand low in-order delivery delays while the high data rates require all the available resources of the MP-MH

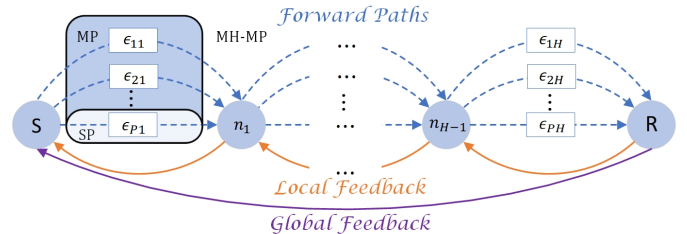


Figure 1. Multi-hop multipath network with delayed feedback. The single-path, multipath, and multi-hop multipath communications are denoted by SP, MP, and MH-MP, respectively.

networks. Traditional information-theory solutions with large blocklength are not able to reach this desired trade-off.

In point-to-point channels with erasures and delayed feedback, different packet-level techniques have been considered to manage and reduce the effects of this throughput-delay trade-off. Using forward error correction (FEC) according to the feedback acknowledgments, the in-order delivery delay can be reduced [3], and the performance of SR-ARQ (Selective Repeat - Automatic Repeat reQuest) protocols can be boosted [4]. Moreover, coding solutions are considered as well [5], [6]. However, although part of those solutions are reactive to the feedback acknowledgments (i.e. causal), none of those solutions are tracking the varying channel condition and the rate (i.e. not adaptive). In [7] for SP communication, we have recently provided a novel adaptive and causal random linear network coding (AC-RLNC) solution with FEC. To ensure low in-order delivery delays, in this solution the sender tracks the channel variations (i.e. adaptive) using a priori and posteriori algorithms (i.e. causal), and adapts the code rate according to the erasure realizations.

Although joint optimization of coding and scheduling in SP communications has been considered in [7], joint coding and scheduling in MP and MH networks for optimizing the trade-off between throughput and in-order delivery delay remains a challenging open problem. To generalize the AC-RLNC solution to MP networks, the main difficulty lies in the discrete allocation of the new coded packets of information and FEC coded packets, over the available paths. On one hand, in order to achieve a high throughput, the sender needs to send as many packets of information as possible. On the other hand, it has to balance the failed transmissions by retransmitting several times the same packets. Furthermore, the earlier the retransmissions are sent, the lower is the delay. To obtain the desired throughput-delay trade-off, the raised open problem is how to allocate transmitted packets to the paths, at each time slot. Moreover, in a MH network, the rate is limited by

the link with the smallest rate, therefore causing bottleneck effects if the channel rate varies from hop to hop. In this paper, we propose a MP and MH adaptive causal network coding solution that can learn the erasure pattern of the channels, and adaptively adjust the allocation of its retransmitted packets across the available paths. Furthermore, in the MH setting, we propose a decentralized adaptive solution which may completely avoid the bottleneck effect by reorganizing the order of global paths at each intermediate node. This novel approach will close the mean and max in-order-delay gap and boost the throughput.

Contributions

We propose a novel adaptive causal coding solution with FEC for MP and MH communications with delayed feedback. The proposed solution generalizes the SP setting, in which the AC-RLNC algorithm can track the network condition, and adaptively adjust its retransmission rates a priori and posteriori over all the available resources in the network. Specifically, to balance the allocation of packets across different paths, the adaptive algorithm we utilize is based on a discrete water-filling approach. Furthermore, in the MH setting, to reduce the bottleneck effect due to the channel variations at each hop, we propose a new decentralized optimization balancing algorithm. This adaptive algorithm can be applied independently at each intermediate node according to the feedback acknowledgments, while tracking the erasure patterns of the income and outcome channels.

We contrast the performance of the proposed approach with the one of SR-ARQ. We demonstrate that the proposed approach can, compared to SR-ARQ, achieve a throughput up to two times better and reduce the delay by a factor of more than three.

The structure of this work is as follows. In Section II, we formally describe the system model and the metrics in use. In Section III, we provide a background on the SP solution and on the MH transmission protocol. In Section IV, we present the MP solution with the simulation results. In Section V, we generalize the MP solution to MP-MH solution. Finally, we conclude the paper in Section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a real-time slotted communication model with feedback in two different settings: first a multipath (MP) channel with P paths and then a multipath and multi-hop (MH) channel with H hops and P paths per hop, as represented in Figure 1. Symbol definitions are provided in Table I.

Multipath setting: At each time slot t , the sender transmits over each path $p \in \{1, \dots, P\}$ a coded packet $c_{t,p}$.¹ Forward paths between the sender and the receiver are modeled as independent binary erasure channels (BEC). Namely, erasure events are i.i.d. with probability ϵ_p for each p -th path. According to the erasure realizations, the receiver sends at each time slot either an acknowledgment (ACK) or a negative-acknowledgment (NACK) message to the sender, using the

¹A negligible size header containing transmission information may be sent with the coded packets.

Parameter	Definition
P, H	number of paths and number of hops
$\epsilon_{p,h}$	erasure probability of the p -th path and h -th hop
$r_{p,h}$	$1 - \epsilon_{p,h}$, rate of the p -th path and h -th hop
t	time slot index
M	number of information packets
$x_i, i \in [1, M]$	information packets
$\mu_i \in \mathbb{F}_z$	random coefficients
$c_{t,p}$	RLNC to transmit at time slot t on path p
D_{mean}, D_{max}	mean and max in-order delivery delay
η	throughput
$k = RTT - 1$	number of new information packets in window
EW	end window of k new packets
$\bar{o}$	maximum window size
m_p	number of FEC's for the p th path
ad	number of added DoF's (global denoted by g)
md	number of missing DoF's (global denoted by g)
d	md/ad , DoF rate
th	retransmission parameter
Δ	$P \cdot (d - 1 - th)$, DoF rate gap

Table I
SYMBOL DEFINITION

same paths. For simplicity, feedback messages are assumed to be reliable, i.e., without errors in the feedback

The delay between the transmission of a coded packet and the reception of the corresponding feedback is called round trip time (RTT). Defining ρ_p as the rate of the forward path p , in bits/second and $|c_{t,p}|$ as the size of coded packet $c_{t,p}$, in bits, the maximum duration of a transmission is $t_d = \max_{t,p} |c_{t,p}| / \rho_p$. Letting t_p be the propagation time between sender and receiver in seconds and assuming the size of feedback messages is negligible compared to the one of coded packets, the RTT is given by

$$RTT = t_d + 2t_p. \quad (1)$$

Hence, for each transmitted coded packet $c_{t,p}$, the sender receives a ACK(t, p) or NACK(t, p) after RTT seconds.

Multipath Multi-hop setting: In this setting, the sender and receiver behave exactly as in the single-hop. For simplicity, we assume that there are P paths in each hop $h \in \{1, \dots, H\}$, each with i.i.d erasure probabilities $\epsilon_{p,h}$. At each time slot, each intermediate node n_h receives from the h -th hop (and therefore either from the sender for the first node or from the previous node for the others) P coded packets from the independent paths. The node then sends P (possibly different) coded packets on the $(h+1)$ -th hop (towards next node or the receiver for the last intermediate node). For feedback acknowledgments, either a local hop-by-hop mechanism (from node to node) or a global one (directly from the receiver to the sender) can be used. Letting $t_{h,p}$ be the propagation delay of one hop, in seconds, and assuming all hops have the same propagation delay, the propagation time t_p becomes $t_p = Ht_{p,h}$.

Our goal for both of these settings, with parameters ρ -th and RTT, is to maximize the throughput, η , while minimizing the in-order delay, D , as given in the following definitions.

Definition 1. Throughput, η . This is the total amount of information (in bits/second) delivered, in order, at the receiver in n transmissions over the forward channel. The normalized throughput is the total amount of information delivered, in order at the receiver divided by n and the size of the packets.

Definition 2. In-order delivery delay, D . This is the difference between the time an information packet is first transmitted in a coded packet by the sender and the time that the same information packet is decoded, in order at the receiver, and successfully acknowledged [1].

More precisely, we consider the mean and max in-order delay D_{mean} and D_{max} . While D_{mean} reduces the overall delay, D_{max} is critical for real-time applications that need a low inter-arrival time between packets.

III. BACKGROUND

Before we consider the MP and MH protocols in details, we present the SP AC-RLNC algorithm suggested in [7] and the capacity achieving protocol described in [8].

a) SP AC-RLNC algorithm: In [7], an adaptive causal random linear network coding algorithm for a single-hop single-path setting is described. We review here several key features of that protocol.

Coded packets and sliding window mechanism: Raw packets of information are encoded using RLNC. Each coded packet c_t , called a degree of freedom (DoF), is obtained as

$$c_t = \sum_{i=w_{min}}^{w_{max}} \mu_i \cdot x_i, \quad (2)$$

with $\{\mu_i\}_{i=w_{min}}^{w_{max}}$, random coefficients in the field $\mathbb{F}_z$ of size z , $\{x_i\}_{i=w_{min}}^{w_{max}}$, the raw information packets and w_{min} and w_{max} , the limits of the current window². At each time step, the sender can either transmit a new coded packet or repeat the last sent combination³. w_{max} is thus incremented each time a new DoF is sent while w_{min} corresponds to the oldest raw packet that is not yet decoded. We denote by $\text{DoF}(c_t)$ the DoF's contained in c_t , i.e. the number of information packets in c_t .

Tracking the path rate and DoF rate: Given the feedback acknowledgments, the sender can track the erasure probability ϵ (and thus the path rate r , defined as $r = 1 - \epsilon$) as well as the DoF rate $d = md/ad$, with md and ad being respectively the number of erased and repeated DoF's. These quantities are needed by the two FEC mechanisms that counteract the erasures.

A priori mechanism: When $k = RTT - 1$ new packets of information have been transmitted, $\lceil \epsilon \cdot k \rceil$ repetitions⁴ of the same RLNC are sent in order to compensate for the expected erasures.

A posteriori mechanism: When a-priori repetitions are not sufficient, the retransmission criterion $r - d \leq th$, with th being a tunable parameter, determines when additional FEC's, called feedback FEC's (FB-FEC) are needed. Intuitively, when the DoF rate is higher (resp. lower) than the rate of the channel, then too many (resp. enough) coded packets are erased, and retransmissions are (resp. are not) needed.

²As shown in [2], if z is large enough, a generation of k raw information packets can be decoded with high probability using Gaussian elimination on k coded packets.

³“Same” and “new” refer here to the raw packets of information contained in the linear combination. Sending the same linear combination thus means that the raw packets are the same but with different random coefficients.

⁴ $\lceil x \rceil$ corresponds to rounding x to the nearest integer.

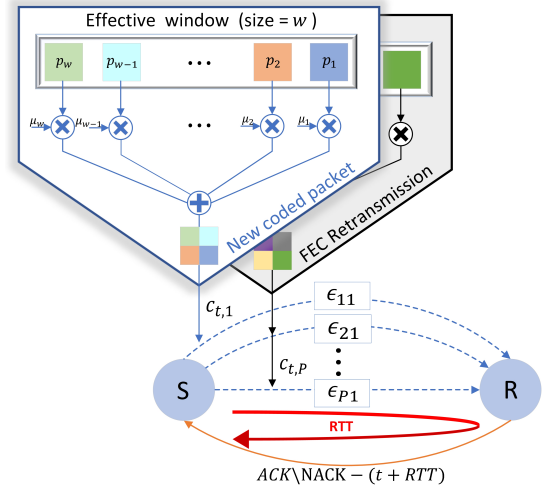


Figure 2. System model and encoding process of the coded RLNC combinations in MP network. In this example, for simplicity of notation, $w_{min} = 1$.

Size limit mechanism: In order to reduce D_{max} , the window size $w = w_{max} - w_{min} + 1$ is limited to $\bar{o}$. Like th , $\bar{o}$ can also be tuned to achieve different trade-offs. When that limit is reached, the sender transmits the same RLNC until it knows, as result of the feedback acknowledgments, that all information packets are decoded.

b) Multi-hop transmission: As proved in [9], the capacity of a network equals its maximum flow (or equivalently its minimum cut). Using random linear network coding, that capacity can be achieved for instance by using the protocol suggested in [8]. In this protocol, each node generates and sends random linear combinations of all packets in its memory whatever the structure of the network. Specifically to the MP and MH setting, at each time slot, each node (the sender included) stores received RLNC's (or the previously sent RLNC's for the encoder). Then, each node sends on the P next paths a linear combination of all received RLNC's. In the asymptotic regime, that protocol achieves the capacity [8].

IV. MULTIPATH COMMUNICATION

In this section, we propose to merge the AC-RLNC solution described in Section III (for SP), with an adaptive algorithm balancing the allocation of new RLNC's and FEC RLNC's on the paths. Figure 2 shows the adaptive causal coding process on the single hop multipath network. The adaptive algorithm is based on a discrete water filling approach, i.e., bit-filling (BF), as given in [10]. However, the BF is modified in order to take into account both rate and in-order delay objectives. To reach the desired trade-off between throughput and delay in the MP network, we suggest to utilize the key features of the SP AC-RLNC algorithm, especially the a-priori and a-posteriori FEC mechanisms, as well as the tracking of the channel rate and the DoF rate via the feedback acknowledgments. Yet, in the MP network, to maximize the throughput while minimizing the in-order delay, allocation of new coded packets and retransmissions demands to consider adaptively the available resources across all the channels. The main components of the packet level protocol and the balanced allocation algorithm over the different paths are described next in Section IV-A. In Section IV-B, the simulation results of the solution we suggest for the MP network are presented.

A. Adaptive Coding Algorithm

Here we detail the MP solution, described in Algorithm 1.

a) *A priori mechanism (FEC)*: After the transmission of $k = P(RTT - 1)$ new RLNC's, $m_p = \lceil \epsilon_p(RTT - 1) \rceil$ FEC's are sent on the p -th path. This mechanism tries to provide a sufficient number of DoF's to the receiver, by balancing the expected number of erasures. Note that m_p may vary from path to path, according to the estimated erasure probability of each path.

b) *A posteriori mechanism (FB-FEC)*: The retransmission criterion of the FB-FEC mechanism has to reflect, at the sender's best knowledge, the ability of the receiver to decode RLNC's. Let md_g be the number of missing DoF's (i.e. the number of new coded packets that have been erased) and ad_g be the number of added DoF's (i.e. the number of repeated RLNC's that have reached the receiver). Due to the RTT delay, md_g and ad_g cannot be computed exactly. We leave the discussion of the possible strategies to calculate md_g and ad_g to future work, because of space limitations. However, one of the possible options is to add DoF's according to the estimated rate of each path. Now defining the DoF rate of MP network as $d = \sum_{p=1}^P md_{g,p} / \sum_{p=1}^P ad_{g,p}$, according to the feedback acknowledgments, and using a tunable parameter th , the retransmission criterion can be rewritten as $d - 1 > th$. Finally, defining the DoF rate gap as, $\Delta = P \cdot (d - 1 - th)$, the FB-FEC criterion we suggest is

$$\text{FB-FEC: } retransmission \iff \Delta > 0. \quad (3)$$

c) *Packet allocation*: In order for the sender to decide on which paths allocate the new coded packets of information and retransmissions of FB-FEC's, we propose a new algorithm inspired by a BF algorithm given in [10]. However, unlike the optimization problem considered in [10], in which there is one objective in order to optimally allocate bits under power constraints, the optimization problem in this paper contains two objectives. On one hand, the throughput needs to be maximized through the allocation of new coded packets while on the other hand, the in-order delay has to be reduced through FB-FEC retransmissions. Figure 3 illustrates the BF packets allocation for a multipath network.

We define the set of all the paths as $\mathcal{P}$, and the index of all the possible sub-sets as $w \in \{1, \dots, 2^P\}$. We denote the possible subset of paths over which the sender will transmit the new coded packets of information as $\mathcal{P}_w$, and the possible subsets of paths over which the sender will transmit the retransmissions of FB-FEC packets as $\mathcal{P}_w^c$.

Proposition 1 (Bit-Filling). *Given the estimated rates of all the paths, $r_p \in \{1, \dots, P\}$, the sender wants to maximize the throughput of the new packets of information, such that*

$$\begin{aligned} \arg \max_{(\mathcal{P}_w)} \quad & \sum_{i \in \mathcal{P}_w} r_i, \\ \text{s.t.} \quad & \sum_{j \in \mathcal{P}_w^c} r_j \geq \Delta \quad \text{for } \mathcal{P}_w^c = \mathcal{P} \setminus \mathcal{P}_w, \end{aligned} \quad (4)$$

where the optimization problem minimizes the in-order delivery delay, by providing over the selected paths a sufficient number of DoF's for decoding.

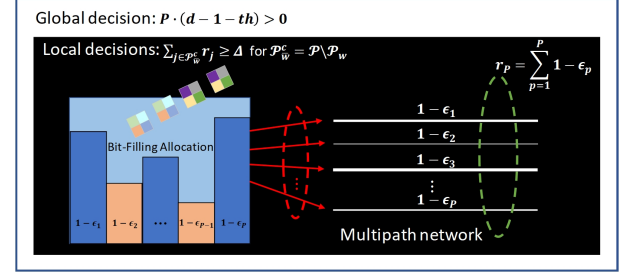


Figure 3. Bit-Filling packets allocation in multipath network. At the sender, given the estimated rates (green), according to the retransmission criterion given in (3), first a global decision is made (i.e., check if retransmission is needed). Then if retransmission is needed, a local decision on which paths to send new packets of information and retransmissions (described by different colors as in Figure 2) are made according to the bit-filling given in Proposition 1 (red).

It is important to note that by tuning the chosen parameter th (reflected in Δ (3)) it is possible to obtain the desired throughput-delay trade-off. Moreover, to maximize the performance of the proposed approach, it is required to solve problem (4) only when the estimations of the rates change. To reduce the complexity of the optimization problem, once the number of paths is high, we can consider a relaxation of the optimization, e.g., using Knapsack problem algorithms [11].

Algorithm 1 multipath protocol for packet scheduling

```

1: Initialize transmission:
2: while packets to transmit do
3:   if Feedback available then
4:     Update  $\epsilon_p$  for each path
5:     Update  $md_g$  and  $ad_g$ 
6:     Update  $\Delta$ 
7:   end if
8:   Size limit transmissions:
9:   if  $w > \bar{o}$  then
10:    Retransmit same RLNC until  $\text{DoF}(c_t) = 0$ 
11:   else
12:     FEC transmissions:
13:     for all paths with  $m_p > 0$  do
14:       Retransmit same RLNC
15:        $m_p = m_p - 1$ 
16:     end for
17:     if remaining paths6 then
18:       FB-FEC transmissions:
19:       if  $\Delta > 0$  then
20:         Determine FB-FEC paths
21:         Transmit same RLNC on these paths
22:       end if
23:       New transmissions:
24:       for all remaining paths do
25:         if not EW then
26:           Transmit new RLNC
27:         end if
28:       end for
29:       FEC transmission (initialization):
30:       if EW then
31:         Set  $m_p := \lceil \epsilon_p(RTT - 1) \rceil$ 
32:         for all remaining paths do
33:           Transmit same RLNC
34:            $m_p = m_p - 1$ 
35:         end for
36:       end if
37:     end if
38:   end if
39: end while

```

B. Simulation Results

The performance of the MP AC-RLNC protocol is compared with two other protocols, as presented in Figure 4.

⁶The paths that have not yet an assigned RLNC for that time step.

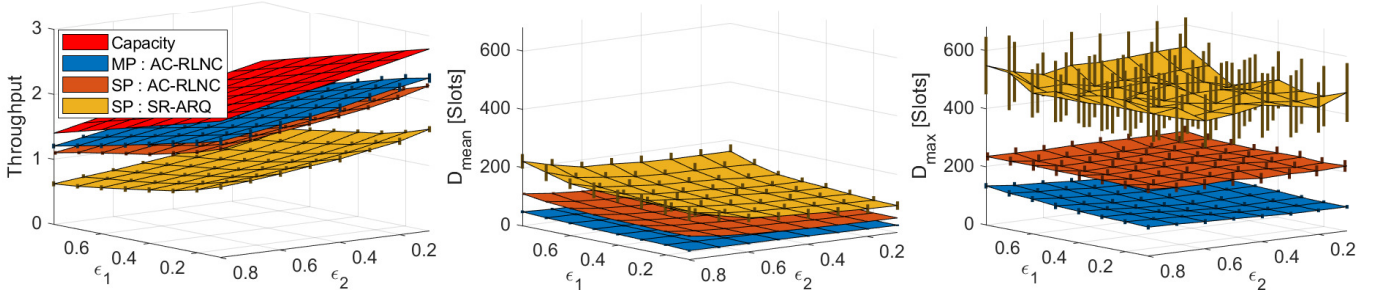


Figure 4. Performances of multipath protocol with $RTT = 20$ [slots], $th = 0$ and $\bar{o} = 2k$, on 4 paths with $\epsilon_3 = 0.2$ and $\epsilon_4 = 0.8$, averaged on 150 iterations.

a) Setting and protocols: We consider the setting of Figure 1, with $H = 1$, $P = 4$, $RTT = 20$, and with $\epsilon_{31} = 0.2$ and $\epsilon_{41} = 0.8$, while the erasure probabilities of the two other paths (ϵ_{11} and ϵ_{21}) vary in the range of $[0.1 \ 0.8]$.

The MP AC-RLNC protocol has been simulated with $th = 0$ and $\bar{o} = 2w$. To emphasize the gain we get by tracking the channel condition, adjusting the retransmissions, and with the discrete BF algorithm, we compare the MP protocol with the SP AC-RLNC [7], applied independently on each path. The tunable parameters of the SP AC-RLNC protocol are the same as the MP one. Furthermore, we compare these protocols with SR-ARQ, to show the gain we get with adaptive and causal network coding. Again, the SR-ARQ protocol is used independently on each path. The results of Figure 4 have been averaged on 150 different channel realizations, where the filled curves correspond to the mean performances while the error bars represent the standard deviation from the mean.

b) Results: Based on the results of Figure 4, the throughput is slightly increased (around 15%) compared to the SP AC-RLNC protocol but nearly doubled with regards to SR-ARQ, independently of the erasure probabilities. On the delay point of view, both the mean and max in-order delay are dramatically reduced with the MP protocol. More precisely, compared to SP AC-RLNC protocol, the MP protocol performs nearly twice as better in term of mean delay and between 2 to 3 times better in terms of max delay. With regards to the SR-ARQ protocol, performances are nearly 4 times better for the mean delay and between 4 to 6 times better for the max one. For higher RTT 's, the gap between our solution and other protocols increases⁵.

V. MULTI-HOP MULTIPATH COMMUNICATION

In this section, we generalize the MP solution given in Section IV to the MP and MH setting introduced in Section II, in which each node can estimate the erasure probability of the incoming and outgoing paths according to the local feedback. To present our MH protocol, we use the example of Figure 5.

In that MH setting, in the asymptotic regime, using RLNC, the min-cut max-flow capacity C (2.6 in the example) can be achieved by mixing together coded packets from all the paths at each intermediate node (see Section III). Hence, one could use P parallel SP AC-RLNC protocols with the node recoding protocol to get a throughput very close to the min-cut max-flow capacity. However, due to the mixing between the paths, dependencies are introduced between the FEC's and the new RLNC's. This will thus result in a high in-order delay.

⁵Those results are not shown here due to limited space.

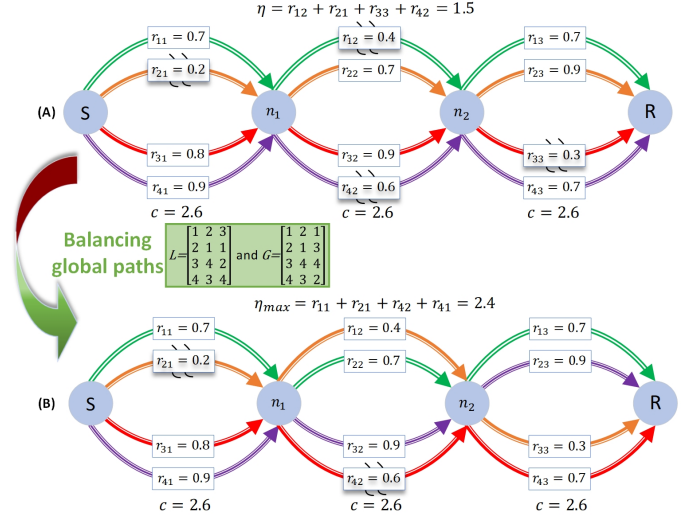


Figure 5. Balancing optimization and global paths example. The numbers correspond to the rates of the paths.

To reduce the in-order delay, the MP algorithm we suggest in Section IV can be used on P global paths, using RLNC independently on each path. A naive choice of global paths is shown in the upper part of Figure 5⁶. In that setting, due to the min-cut max-flow capacity, the maximum throughput of each path is limited by its bottleneck⁷ (i.e. the link with the smallest rate). Doing so, the maximum throughput (i.e. the sum of the min-cut of each path) can thus be much lower than the capacity of the network, as it can be seen from the example of Figure 5, in which the throughput is 1.5.

With these two previous attempts minded, we finally suggest to determine the global paths using a decentralized balancing algorithm whose aim is to maximize the maximal throughput of the network. The second part of Figure 5 shows the global paths resulting from that balancing, and as a result a maximum throughput equal to 2.4. Moreover, it can be seen that only 2 global paths are now affected by the bottleneck links. Once these global paths are defined, the MP AC-RLNC protocol can be used as described in Algorithm 1.

In section V-A, the definition of the global paths is described as well as the full multi-hop algorithm, while the simulation results of the protocol are shown in Section V-B.

A. Adaptive Coding Algorithm

Here we describe the suggested MP and MH solution.

⁶The paths are described by their color and their type of arrow.

⁷In Figure 5, bottlenecks are denoted by a curly symbol behind the rate.

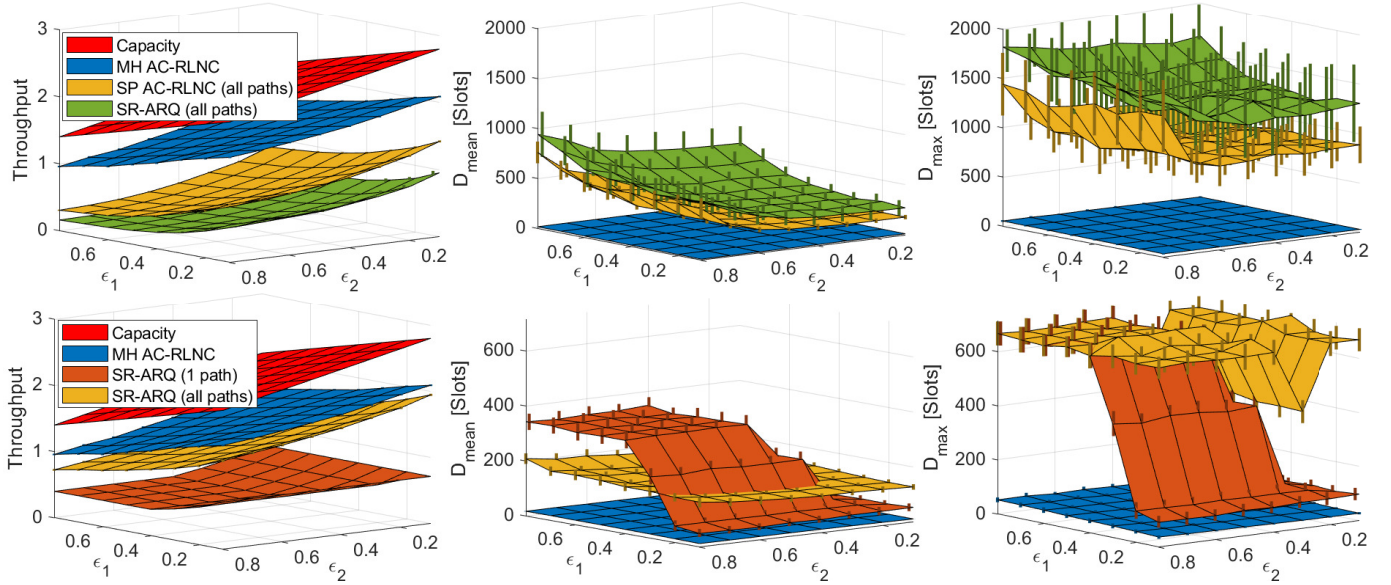


Figure 6. Performances of the multi-hop protocol with $RTT = 12$ [slots], $th = 0$ and $\bar{o} = 2k$, on 4 paths, averaged on 150 iterations. The top results are for end-to-end feedback (from the receiver to the sender), while the bottom results are for hop-by-hop feedback (from each intermediate node and receiver).

a) *Global paths - problem formulation*: In order for the h -th node to transmit packets over the paths maximizing the rate, it needs to know the local matching $L(p, h)$, such that, $L(p, h) = j$ implies that the j -th path of the $(h + 1)$ -th hop is matched with the p -th path of the h -th hop. The definition of the global paths can be done equivalently through a global matching $G(p, h)$, such that, $G(p, h) = j$ implies that the j -th path of the h -th hop belongs to the p -th global path⁹. We point out that even if these two definitions are equivalent, the local matching is particularly convenient to express the global paths in a decentralized way. Moreover, it is important to note that, for L and G to be an admissible matching, each local path must be matched with exactly one other local path at each node. The values of L and G are shown in the example given in Figure 5.

Once admissible global paths are determined, the maximum achievable throughput η_{max} corresponds to the sum of the min-cut of each global path. Defining $r_{G(p,h)h}$ as the rate of the $G(p, h)$ -th path in the h -th hop, η_{max} is thus equal to

$$\eta_{max}(G) = \sum_{p=1}^P \min_{h=1 \dots H} r_{G(p,h)h}. \quad (5)$$

b) *Global paths - suggested solution*: Looking at each global path independently, it appears that rate is lost each time two segments of the global paths are unbalanced (i.e. having different rates). In order to match the paths in a balanced way, we thus suggest the following optimization problem.

Proposition 2 (Balancing Optimization). *Given incoming paths with rate r_{in} and outgoing paths with rate r_{out} , the matching l between these paths is defined as*

$$l = \arg \min_{\tilde{l} \in \text{Perm}(P)} \sum_{p=1}^P |r_{in}(p) - r_{out}(\tilde{l}(p))|, \quad (6)$$

with $r_{out}(\tilde{l}(p))$ the rate of the $\tilde{l}(p)$ -th outgoing path and $\text{Perm}(P)$ being the set of all permutations¹⁰ of the vector $[1, 2, \dots, P]$.

Using (6), the matching of the MP-MH channel is defined in the following way. The first node finds the first matching l_1 by solving (6) with $r_{in}(p) = r_{p1}$ and $r_{out}(p) = r_{p2}$. Then, for the second node, the incoming rate r_{in} can be computed as the min-cut of the partially built paths and r_{out} is set to the rates of next hop. Again, (6) can be solved to find l_2 . This procedure can be repeated until the last node is reached and the local matching L is built as $L = [l_1, l_2, \dots, l_{H-1}]$. Algorithm 2 summarizes the protocol.

Algorithm 2 Greedy balancing protocol

- 1: Set $r_{in}^{(1)}(p) = r_{p1}$, $\forall p = 1 \dots P$
 - 2: **for** $h = 1 \dots H - 1$ **do**
 - 3: Set $r_{out}^{(h)}(p) = r_{p,h+1}$, $\forall p = 1 \dots P$
 - 4: Find l_h by solving (6)
 - 5: Permute $r_{in}^{(h)}$ according to l_h to get r^* , such that $r^*(p)$ is matched with $r_{p,h+1}$
 - 6: Set $r_{in}^{(h+1)}(p) = \min(r^*, r_{p,h+1})$
 - 7: Send $r_{in}^{(h+1)}$ to the next node
 - 8: **end for**
-

Note that Algorithm 2 can be implemented in a decentralized way, with a minimum amount of communication between the nodes. Since the nodes can estimate the erasure probabilities of the adjacent paths, the only needed communication at the h -th hop is the propagation of $r_{in}^{(h)}$.

c) *Global paths - Efficient balancing*: In order to determine the matching in an efficient way, (6) can be reformulated as a minimum weight matching in a bipartite graph, also called *assignment problem* [12]. Indeed, a complete bipartite graph with two classes having P vertices each, corresponding respectively to r_{in} and r_{out} , can be built. The edge linking $r_{in}(p)$ and $r_{out}(p^*)$ is weighted by $|r_{in}(p) - r_{out}(p^*)|$. In that graph, (6) is equivalent to finding a perfect matching (each input path is matched with exactly one output path) with a

⁹The global matching of the first hop is such that the p -th local path belongs to the p -th global path, i.e. $G(p, 1) = p$, $\forall p = 1 \dots P$

¹⁰This restriction prevents non-admissible matchings.

minimum weight (minimize the sum of the rate differences). Several efficient solutions are suggested in the literature for balancing. For example, we can utilize the so called Hungarian algorithm [12], its complexity being in $\mathcal{O}(n^3)$.

d) *Selective mixing*: The AC-RLNC MP protocol can be applied on the balanced global paths. Yet, mixing packets between some of the paths can improve the algorithm. From Figure 5, it can be seen on one hand that defining global paths reduces the maximum throughput. On the other hand, mixing all packets between all the paths suppresses the usefulness of FEC and FB-FEC transmissions. But if at intermediate nodes, new packets are mixed together on one hand and the FEC's and FB-FEC on the other hand, the throughput will be increased without increasing the delay.

B. Simulation Results

The performances of the MP-MH AC-RLNC protocol are compared with two other protocols, as presented in Figure 6.

a) *Setting and protocols*: We consider the setting of Figure 1, with $H = 3$, $P = 4$, and $RTT = 12$. Where $\epsilon_{21} = 0.8$, $\epsilon_{31} = 0.2$, $\epsilon_{12} = 0.6$, $\epsilon_{42} = 0.4$, $\epsilon_{13} = 0.3$, $\epsilon_{33} = 0.7$, while the erasure probabilities of the other paths are equal to $\epsilon_{11} = \epsilon_{22} = \epsilon_{13} := \epsilon_1$ and $\epsilon_{41} = \epsilon_{32} = \epsilon_{43} := \epsilon_2$ with ϵ_1 and ϵ_2 varying in the range of $[0.1 \ 0.8]$.

The proposed protocol has been simulated with $th = 0$ and $\bar{o} = 2w$. First, we compare in the upper graph of Figure 6 our solution with end to end protocols. Specifically, as for the multipath case, the SP AC-RLNC protocol and the SR-ARQ protocol are applied independently on the P balanced global paths we get with Algorithm 2. However, no recoding is performed at the intermediate nodes. Secondly, in order to not depend on node recoding, we investigate in the lower graph of Figure 6 the performances of the hop by hop SR-ARQ protocol on two different settings. In one setting, it is applied to one single global path that is build from the best path of each hop. In the other, it is applied on the P balanced global paths.

The metrics have been averaged on 150 different channel realizations, where the filled curves correspond to the mean performances while the error bars represent the standard deviation, as for the MP results.

b) *Results*: From Figure 6, one can see that the MP-MH AC-RLNC protocol performs dramatically better both with regards to the rate and in-order delay. As expected, in the upper graph, one can see that the rate is improved a lot with regards to end to end protocols. The rate is doubled for good channel conditions ($\epsilon_1 = \epsilon_2 = 0.1$) and multiplied by 3 for bad channel conditions ($\epsilon_1 = \epsilon_2 = 0.8$), for the SP AC-RLNC algorithm. Compared with the SP SR-ARQ protocols, performances are even better. Yet, the improvements are even more dramatic from the delay point of view. The mean and max in-order delay are reduced by a factor 15 for good channel conditions and up to a factor 40 for bad ones, for both the AC-RLNC and the SR-ARQ protocols. From the lower part of Figure 6, it can be seen that the hop by hop SR-ARQ protocol is also much worse than our solution. The improvement we get on the rate is small (10%) for good channel conditions but it becomes significant (35%) when channels are bad. From

the delay point of view, the gain is significant since both the mean and max in-order delay are reduced approximately by a factor 10. The hop by hop SR-ARQ protocol on only 1 path has obviously a much lower rate. For the in-order delay, the improvements we get highly depend on the channel configuration (changing slightly the erasure probabilities gives very different results) but the MH AC-RLNC protocol is still better independently of the configuration of the channel.

VI. CONCLUSION

We proposed a MP-MH adaptive and causal coding algorithm for communications with delayed feedback. The MP algorithm, and especially the combination of the a-priori and a-posteriori mechanisms with efficient bit-filling packet allocation, outperforms significantly the existing protocols. By tuning the parameters of the algorithm, the desired throughput-delay trade-off is obtained. Splitting the MH protocol into the suggested balancing optimization and the MP algorithm gives very promising results. Indeed, MP-MH AC-RLNC protocol has a higher throughput and a much lower delay compared to SR-ARQ. Specifically, in the end-to-end setting without recoding in the intermediate nodes, it reaches a very high delay and a lower throughput. The hop-by-hop SR-ARQ is less impacted by the absence of recoding. Nevertheless, this comes at the price of a high sensitivity to the channel configuration and the order of the hops, while each node needs to be able to perform the full SR-ARQ protocol.

Future work includes the derivation of bounds on the mean and maximum in order delivery delay and throughput. Extensions also include the study of general mesh networks.

ACKNOWLEDGMENT

We want to thank Maxence Vigneras for contributing in the MATLAB implementation of the suggested solution.

REFERENCES

- [1] W. Zeng, C. T. Ng, and M. Médard, "Joint coding and scheduling optimization in wireless systems with varying delay sensitivities," in *Proc., IEEE Commun. Soc. Conf. Sensor, Mesh and Ad Hoc Commun. and Netw.*, 2012, pp. 416–424.
- [2] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [3] M. Karzand and D. J. Leith, "Low delay random linear coding over a stream," in *Proc., IEEE Allerton Conf. Commun., Control, and Computing*, Sep. 2014, pp. 521–528.
- [4] J. Cloud, D. Leith, and M. Médard, "A coded generalization of selective repeat ARQ," in *2015 IEEE Conf. Comp Commun.*, 2015.
- [5] G. Joshi, Y. Kochman, and G. W. Wornell, "On playback delay in streaming communication," in *Proc. IEEE Int. Sym. Inf. Theory*, 2012.
- [6] D. Malak, M. Médard, and E. M. Yeh, "Tiny codes for guaranteeable delay," *IEEE J. Sel. Areas in Commun.*, vol. 37, pp. 809–825, 2019.
- [7] A. Cohen, D. Malak, V. B. Bracha, and M. Médard, "Adaptive causal network coding with feedback," *arXiv preprint arXiv:1905.02870*, 2019.
- [8] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Commun.*, vol. 1, pp. 3–20, Mar. 2008.
- [9] S. Li, R. Yeung, and N. Cai, "Linear network coding," *Inf. Theory, IEEE Trans. on*, vol. 49, pp. 371 – 381, 03 2003.
- [10] N. Papandreou and T. Antonakopoulos, "Bit and power allocation in constrained multicarrier systems: The single-user case," *EURASIP Jour. Adv. Sig. Process.*, vol. 2008, no. 1, p. 643081, 2007.
- [11] M. Laurie, "An algorithm for the 0/1 knapsack problem," *Mathematical Programming*, vol. 14, no. 1, pp. 1–10, Dec 1978. [Online]. Available: <https://doi.org/10.1007/BF01588947>
- [12] M. J., "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.