

# Dynamic Control of Coding for Progressive Packet Arrivals in DTNs

Eitan Altman, Lucile Sassatelli, Francesco de Pellegrini

► **To cite this version:**

Eitan Altman, Lucile Sassatelli, Francesco de Pellegrini. Dynamic Control of Coding for Progressive Packet Arrivals in DTNs. IEEE Transactions on Wireless Communications, Institute of Electrical and Electronics Engineers, 2013, 12 (2), pp.725-735. 10.1109/TWC.2012.122212.120196 . hal-00917413

**HAL Id: hal-00917413**

**<https://hal.inria.fr/hal-00917413>**

Submitted on 11 Dec 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Dynamic control of Coding for general packet arrivals in DTNs

Eitan Altman, Francesco De Pellegrini and Lucile Sassatelli

## Abstract

Delay tolerant Networks (DTNs) leverage the mobility of relay nodes to compensate for lack of persistent connectivity. In order to decrease message delivery delay, the information to be transmitted can be replicated in the network. For general packet arrivals at the source and two-hop routing, we derive performance analysis of replication-based routing policies and study their optimization. In particular, we find out the conditions for optimality in terms of probability of successful delivery and mean delay and devise optimal policies, so-called *piecewise threshold policies*. We account for linear block-codes as well as rateless random linear coding to efficiently generate redundancy, as well as for an energy constraint in the optimization. We numerically assess the higher efficiency of piecewise threshold policies compared with other policies by developing heuristic optimization of the thresholds for all flavors of coding considered.

## Index Terms

Delay Tolerant Networks, Mobile Ad Hoc Networks, Optimal Scheduling, Rateless codes, Network coding

## I. INTRODUCTION

Delay Tolerant Networks (DTNs) leverage contacts between mobile nodes and sustain end-to-end communication even between nodes that do not have end-to-end connectivity at any given instant. In this context, contacts between DTN nodes may be rare, for instance due to low densities of active nodes, so that the design of routing strategies is a core step to permit timely delivery of information to a certain destination with high probability. When mobility is random, i.e., cannot be known beforehand, this is obtained at the cost of many replicas of the original information, a process which consumes energy and memory resources. Since many relay nodes (and thus network resources) may be involved in ensuring successful delivery, it becomes crucial to design efficient resource allocation and data storage protocols. The basic questions are then:

- (i) **transmission policy:** when the source is in contact with a relay node, should it transmit a packet to the relay?

(ii) **scheduling:** if yes, which packet should a source transfer?

In the basic scenario, the source has initially all packets. Under this assumption it was shown in [1] that the transmission policy has a threshold structure: it is optimal to use all opportunities to spread packets till some time  $\sigma$  and then stop. This policy resembles the well-known ‘‘Spray-and-Wait’’ policy [2]. In this work we assume a more general arrival process of packets: they need not to be simultaneously available for transmission initially, i.e., when forwarding starts, as assumed in [1]. This is the case when large multimedia files are recorded at the source node that sends them out without waiting for the whole file reception to be completed.

**Contributions.** This paper focuses on general packet arrivals at the source and two-hop routing. The contributions are fourfold:

- For work-conserving policies, we find out the conditions for optimality in terms of probability of successful delivery and mean delay.
- We prove that work-conserving policies are always outperformed by so-called *piecewise threshold policies*. These policies are the extension of threshold policies derived in [1], [3], for the case of general packet arrival at the source.
- We extend the above analysis to the case where redundant packets are coded packets, generated both with linear block-codes and rateless coding. We also account for an energy constraint in the optimization.
- We illustrate numerically the higher efficiency of piecewise threshold policies compared with work-conserving policies by developing heuristic optimization of the thresholds for all flavors of coding considered.

More in details, assume  $\mathbf{t} = (t_1, \dots, t_K)$  are the arrival times of the  $K$  packets at the source,  $t_1 \leq t_2 \leq \dots \leq t_K$ . Owing to progressive arrivals, work-conserving policies (i.e., when the source sends a packet with probability 1 all the time) can be suboptimal, in contrast to [1], [3]. Defining couples  $(s_i, p_i)$  for  $i = 1, \dots, K$ , where  $s_i$  are thresholds in  $[t_i, t_{i+1}]$  and  $p_i$  are probabilities such that the source sends with probability  $p_i$  only between  $t_i$  and  $s_i$ , we are able to prove analytically that piecewise threshold policies, defined by  $s'_i$  with  $p_i = 1$ , for all  $i = 1, \dots, K$ , always outperform other kind of policies.

**Related Work.** Papers [4] and [5] propose a technique to erasure code a file and distribute the generated code-blocks over a large number of relays in DTNs. The use of erasure codes is meant to increase the efficiency of DTNs under uncertain mobility patterns. In [5] the performance gain of the coding

scheme is compared to simple replication, i.e., when additional copies of the same file are released. The benefit of erasure coding is proved by means of extensive simulations and for different routing protocols, including two hop routing. In [4], the authors address the case of non-uniform encounter patterns, and they demonstrate strong dependence of the optimal successful delivery probability on the way replicas are distributed over different paths. The authors evaluate several allocation techniques; also, the problem is proved to be NP-hard. The paper [6] proposes general network coding techniques for DTNs. In [7] ODE based models are employed under epidemic routing; in that work, semi-analytical numerical results are reported describing the effect of finite buffers and contact times; the authors also propose a prioritization algorithm. The same authors in [8] investigate the use of network coding using the Spray-and-Wait algorithm and analyze the performance in terms of the bandwidth of contacts, the energy constraint and the buffer size. The paper [9] addresses the design of stateless routing protocols based on network coding, under intermittent end-to-end connectivity. A forwarding algorithm based on network coding is specified, and the advantage over plain probabilistic routing is proved when delivering multiple packets.

The structure of the paper is the following. In Sec. II we introduce the network model and the optimization problems tackled in the rest of the paper. Sec. III and Sec. IV describe optimal solutions in the case of work-conserving and not work-conserving forwarding policies, respectively. Sec. V addresses the case of energy constraints. Sec. VI performance analysis of linear block-coding. Rateless coding techniques are presented in Sec. VII. Sec. VIII provides a numerical analysis, and Sec. IX concludes the paper.

## II. THE MODEL

[TABLE 1 about here.]

The main symbols used in the paper are reported in Tab. I. Consider a network that contains  $N + 1$  mobile nodes. We assume that two nodes are able to communicate when they come within reciprocal radio range and communications are bidirectional, and that the duration of such contacts is sufficient to one packet in each direction. Also, let the time between contacts of pairs of nodes be exponentially distributed. The validity of this model been discussed in [10], and its accuracy has been shown for a number of mobility models (Random Walker, Random Direction, Random Waypoint). Following the notation of [11], let  $\beta$  be the intra-meeting intensity, i.e., the mean number of meetings between any two given nodes per unit

of time. Let  $\lambda$  be the inter-meeting intensity (mean number of meetings between one given node and any other nodes within unit time). We have  $\lambda = \beta N$  because we consider the density of nodes is constant, so is  $\lambda$ , when  $N$  increases. That matches to the fact that the number of nodes that fall within radio range per unit of time remains constant if the density of nodes does.

A file is transmitted from a source node to a destination node, and decomposed into  $K$  packets. The source of the file receives the packets at some times  $t_1 \leq t_2 \leq \dots \leq t_K$ .  $t_i$  are called the *arrival times*. We assume that the transmitted file is relevant during some time  $\tau$ , i.e., all the packets should arrive at the destination by time  $t_1 + \tau$ . Furthermore, we do not assume any feedback that allows the source or other mobiles to know whether the file has made it successfully to the destination within time  $\tau$ .

We consider two-hop routing: a packet can go only through one relay. The forwarding policy of the source is as follows. If at time  $t$  the source encounters a mobile which does not have any packet, it gives it packet  $i$  with probability  $u_i(t)$ . Clearly,  $u \leq 1$  where  $u = \sum_i u_i(t)$ . Also, there is an obvious constraint that  $u_i(t) = 0$  for  $t \leq t_i$ .

Let  $\widehat{\mathbf{X}}^{(N)}(t)$  be a  $K$  dimensional vector whose components are  $\widehat{X}_i^{(N)}(t)$ , for  $i = 1, \dots, K$ . Here,  $\widehat{X}_i^{(N)}(t)$  stands for the fraction of mobile nodes (excluding the destination) that have at time  $t$  a copy of packet  $i$ , in a network of size  $N$ . Let  $\widehat{X}^{(N)}(t) = \sum_{i=1}^K \widehat{X}_i^{(N)}(t)$ . This models the spreading process of the  $i$ -th packet in the network: in the following we will refer to fluid approximations that describe the dynamics of the fraction of infected nodes. The validity of such approximation is detailed below. In particular, owing to the dependence of this process on  $\beta$ , and hence on  $N$ , we cannot readily apply the convergence results of [12]. Below we detail the approach based on [13] so as to derive analytical expressions of some performance measures.

#### A. Performance measures: mean-field approximations

Consider time  $t$  sampled over the discrete domain, i.e.,  $t \in \mathbb{N}$ . For our case, the drift defined in [13] is

$$f^{(N,i)}(m) = \mathbb{E} \left( \widehat{X}_i^{(N)}(t+1) - \widehat{X}_i^{(N)}(t) \mid \widehat{X}^{(N)}(t) = m \right) .$$

Owing to the model, we have  $f^{(N,i)}(m) = u_i(t)\beta(1-m)$ . By checking condition **H2** of [13], there exists a function  $\epsilon(N)$  such that  $\lim_{N \rightarrow \infty} \epsilon(N) = 0$  and, for all  $m \in [0, 1]$ ,  $\lim_{N \rightarrow \infty} \frac{f^{(N,i)}(m)}{\epsilon(N)} = f_i(m)$ . Indeed,  $\epsilon(N) = \beta(N)/\lambda$ , fulfills the condition, since  $\lim_{N \rightarrow \infty} \beta = 0$  and  $\lambda$  is a constant in  $N$ . This interaction process is then said to have vanishing intensity [13]. In such cases, Benaïm and Le Boudec have shown

in [13] that, provided that we change the time scale, the re-scaled process  $\tilde{X}_i^{(N)}(r)$  converges for large  $N$ , in mean-square, to a deterministic dynamic system which is the solution of a certain Ordinary Differential Equation (ODE). More precisely,  $\tilde{X}_i^{(N)}(r)$  is defined as a continuous time process by

$$\begin{cases} \tilde{X}_i^{(N)}(t\epsilon(N)) = \hat{X}_i^{(N)}(t) & \text{for all } t \in \mathbb{N} \\ \tilde{X}_i^{(N)}(r) & \text{is affine on } r \in [t\epsilon(N); (t+1)\epsilon(N)] \end{cases}$$

Then  $\tilde{X}_i^{(N)}(r)$  converges to a deterministic process  $\bar{X}_i^{(N)}(r)$  which is the solution of  $\frac{d\bar{X}_i^{(N)}(r)}{dr} = f_i(\bar{X}^{(N)})$ . Let us denote  $\epsilon(N)$  by  $\epsilon$  in what follows for lighter notation. Let us define  $\bar{u}_i(r)$  as the re-scaled version of  $u_i(t)$ , in the same way as for  $\tilde{X}_i^{(N)}(r)$ . The fraction of nodes holding packet  $i$  is approximated, for large  $N$ , by the solution of:  $\frac{d\bar{X}_i^{(N)}(r)}{dr} = \frac{\bar{u}_i(r)\beta}{\epsilon}(1 - \bar{X}_i^{(N)}(r))$ ,  $\bar{X}_i^{(N)}(0) = z$ . We get

$$\bar{X}_i^{(N)}(r) = 1 - (1 - z) \exp\left(-\frac{\beta}{\epsilon} \int_0^r \bar{u}_i(s) ds\right). \quad (1)$$

In general, the term  $\frac{\beta}{\epsilon}$  appearing in (1) does not need to coincide with the inter-meeting intensity  $\lambda$ . With our choice for the scaling law it does, so that, for large values of  $N$ , we obtain the usual expression [11]:  $\bar{X}_i^{(N)}(r) = 1 - (1 - z) \exp(-\lambda \int_0^r \bar{u}_i(s) ds)$ . We can also express the limit  $\bar{P}_i(r)$  of the cumulative distribution function (CDF) of the re-scaled delay for packet  $i$  to make it successfully to its final destination. Then we have:  $\frac{d\bar{P}_i(r)}{dr} = \frac{\beta N}{\epsilon} \bar{X}_i^{(N)}(r)(1 - \bar{P}_i(r))$ , whereby  $\bar{P}_i(r) = 1 - \exp\left(-\frac{\beta N}{\epsilon} \int_0^r \bar{X}_i^{(N)}(s) ds\right)$ . The above expression is an approximation of the re-scaled probability of reception as  $N$  tends to infinity (note that the limit is zero as  $\lim_{N \rightarrow \infty} \beta = 0$ ). As mentioned in [13] (page 15), the above derivation can be used to approximate the random variable  $\hat{X}_i^{(N)}(t)$  by  $X_i^{(N)}(t) = \bar{X}_i^{(N)}(\epsilon t) = 1 - (1 - z) \exp(-\beta \int_0^t u_i(s) ds)$  for large values of  $N$ . As well, we can approximate the CDF of the delivery delay of packet  $i$ , denoted by  $P_i(t)$ , by  $\bar{P}_i(\epsilon t)$ . Then we readily get the approximation of the probability of reception of the  $K$  packets by  $\tau$ :  $P_s(\tau) = \prod_{i=1}^K \bar{P}_i(\epsilon \tau)$ . Note that  $P_i(t) = \bar{P}_i(\epsilon t) = 1 - \exp\left(-\lambda \int_0^t X_i^{(N)}(s) ds\right)$ . In the sequel of the paper, we will use the so-defined notation  $Z_i(\tau) = \int_0^\tau X_i^{(N)}(s) ds$ . We can also get an approximation of the mean completion time for large values of  $N$ :  $\mathbb{E}[D] = \int_0^\infty 1 - P_s(t) dt$ .

## B. Problem statement

We shall now introduce two classes of forwarding policies.

*Definition 2.1:* We define  $u$  to be a *work-conserving* policy if whenever the source meets a node then it forwards it a packet, unless the energy constraint has already been attained.

*Definition 2.2:* We define  $u$  to be a *threshold* policy if the source forwards a packet up to threshold time  $t_s$  and stops forwarding afterwards.

We shall study the following optimization problems:

- **P1.** Find  $\mathbf{u}$  that maximizes the probability of successful delivery till time  $\tau$ .
- **P2.** Find  $\mathbf{u}$  that minimizes the expected delivery time over the work-conserving policies.

An policy  $\mathbf{u}$  is called *uniformly optimal* for problem P1 if it is optimal for problem P1 for all  $\tau > 0$ .

*Remark 2.1:* Note that the forwarding policies we will consider are deployable as the source fully controls the dissemination in a two-hop scheme, provided that information of network parameters such as  $N$  and  $\lambda$  is available at the source node. As showed in [14], direct estimation can even be avoided provided that stochastic approximation algorithms are employed, but, such techniques are beyond the scope of the present paper.

*Energy Constraints:* We can denote by  $\mathcal{E}(t)$  the energy consumed by the whole network for transmitting and receiving the file during the time interval  $[0, t]$ . We adopt a linear model, where  $\mathcal{E}(t)$  is proportional to  $X(t) - X(0)$  because packets are transmitted only to mobiles that do not have any. In particular, let  $\varepsilon > 0$  be the energy spent to forward a packet during a contact, including the energy spent to receive the file at the receiver side. We thus have  $\mathcal{E}(t) = \varepsilon(X(t) - X(0))$ . In the following we will denote  $x$  as the maximum number of packets that can be released due to energy constraints. Accordingly, we introduce constrained problems **CP1** and **CP2** obtained from problems P1 and P2, respectively, by restricting to policies for which the energy consumption till time  $\tau$  is bounded by some positive constant.

### III. OPTIMAL SCHEDULING

#### A. An optimal equalizing solution

*Theorem 3.1:* Fix  $\tau > 0$ . Assume that there exists some policy  $\mathbf{u}$  satisfying  $\sum_{i=1}^K u_t^i = 1$  for all  $t$  and  $\int_0^\tau X_i(t)dt$  is the same for all  $i$ 's. Then  $\mathbf{u}$  is optimal for P1.

**Proof.** Define the function  $\zeta$  over the real numbers:  $\zeta(h) = 1 - \exp(-\lambda h)$ . Denote  $\mathbf{Z} = (Z_1, \dots, Z_K)$  such that  $Z_i = \int_0^\tau X_i(v)dv$ , and let  $Z_{total}$  be  $Z_{total} = \sum_{i=1}^K Z_i$ . We note that  $\zeta(h)$  is concave in  $h$  and that  $\log P_s(\tau, \mathbf{u}) = \sum_{i=1}^K \log(P_i) = \sum_{i=1}^K \log(\zeta(Z_i))$ . It then follows from Jensen's inequality that, for a given fixed  $Z_{total}$ , the success probability when using  $\mathbf{u}$  satisfies  $\log P_s(\tau, \mathbf{u}) \leq K \log(\zeta(Z_{total}/K))$  with equality if  $Z_i$  are the same for all  $i$ 's. Moreover,  $\log(\zeta(Z_{total}/K))$  is increasing with  $Z_{total}$ , and a work-conserving policy achieves the highest feasible  $Z_{total}$  by construction. This implies the Theorem.  $\diamond$

*Note:* This theorem can be also proven by proof of Theorem 6.1 by taking  $H = 0$ .

It is worth noting that the reciprocal of the above theorem is not true: amongst all feasible policies, the policy  $\mathbf{u}$  giving the highest delivery probability is not necessarily WC or equalizing the  $Z_i$ . It is easy to exhibit counter-examples. Explanation is as follows. The limiting parameters for  $Z_i$  are (i) the arrival time of packet  $i$  and (ii) the number of already occupied nodes when packet  $i$  starts to spread.

- The optimal policy for P1 may not be WC:

Let  $Z_{total} = \sum_{i=1}^K Z_i$ . The maximum of  $Z_{total}$  is obtained for a WC policy. For fixed  $Z_{total}$ , the maximum of  $P_s(\tau)$  is indeed obtained by an equalizing policy, but such policy may not belong to the set of feasible policies. In such case, the maximum of  $P_s(\tau)$  may be reached by a non-WC policy (non-maximum  $Z_{total}$ ). For example in the case the last packet arrives much later than the other packets and if  $t_K - \tau$  is high enough, then the network must not be saturated so as to allow the last packet to spread enough until  $\tau$ .

- The optimal policy for P1 may not equalize the  $Z_i$ :

Consider success probability  $P_s(\tau) = \prod_{i=1}^K (1 - \exp(-\lambda Z_i))$  and maximization of  $P_s(\tau)$  over the policies equalizing the  $Z_i$ . Such a policy maximizes the minimum of all  $Z_i$ . Hence, constraining the  $Z_i$  to be equal will set them to the  $\min_i Z_i$ , since  $Z_i$  is anyway constrained by the arrival time of packet  $i$  (e.g., case of  $i = K$  arrived only a few instants before  $\tau$ ). That is why constraining to  $Z_i$  the  $Z_j$  of other packets arrived earlier than packet  $i$  is not optimal: it may be better to decrease a bit  $Z_i$  while increasing a lot  $Z_j$  so that  $P_s(\tau)$  gets increased. For example in the case the last packet arrives much later than the other packets and if  $t_K - \tau$  is low, then the spreading of packet  $K$  will be limited by  $t_K - \tau$  and it is not efficient to limit too much the spreading of the other packets (i.e., lowering the  $Z_i$  for  $i < K$ ) to let more room than needed by packet  $K$ .

## B. Schur convexity majorization

*Definition 3.1:* (Majorization and Schur-Concavity [15])

Consider two  $n$ -dimensional vectors  $d(1), d(2)$ .  $d(2)$  majorizes  $d(1)$ , which we denote by  $d(1) \prec d(2)$ , if

$$\sum_{i=1}^k d_{[i]}(1) \leq \sum_{i=1}^k d_{[i]}(2), \quad k = 1, \dots, n-1, \quad (2)$$

$$\text{and } \sum_{i=1}^n d_{[i]}(1) = \sum_{i=1}^n d_{[i]}(2), \quad (3)$$



where  $d_{[i]}(m)$  is a permutation of  $d_i(m)$  satisfying  $d_{[1]}(m) \geq d_{[2]}(m) \geq \dots \geq d_{[n]}(m)$ ,  $m = 1, 2$ . A function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is Schur concave if  $d(1) \prec d(2)$  implies  $f(d(1)) \geq f(d(2))$ .

Separable Schur concave functions are defined in the following result [15, Proposition C.1 on p. 64].

*Lemma 3.1:* Assume that a function  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  can be written as the sum  $g(d) = \sum_{i=1}^n \psi(d_i)$  where  $\psi$  is a concave function from  $\mathbb{R}$  to  $\mathbb{R}$ . Then  $g$  is Schur concave.

*Theorem 3.2:*  $\log P_s(\tau, \mathbf{u})$  is Schur concave in  $\mathbf{Z} = (Z_1, \dots, Z_K)$ . Hence if  $\mathbf{Z} \prec \mathbf{Z}'$  then  $P_s(\tau, \mathbf{u}) \geq P_s(\tau, \mathbf{u}')$ .

### C. Constructing an optimal work-conserving policy

We propose an algorithm that has the property that it generates a policy  $\mathbf{u}$  which is optimal not just for the given horizon  $\tau$  but also for any horizon shorter than  $\tau$ . Yet optimality here is only claimed with respect to work-conserving policies. We need some auxiliary definitions that we list in order:

- $Z_j(t) := \int_{t_1}^t X_j(r) dr$ . We call  $Z_j(t)$  the cumulative contact intensity (CCI) of class  $j$ .
- $I(t, A) := \min_{j \in A} (Z_j, Z_j > 0)$ . This is the minimum non zero CCI over  $j$  in a set  $A$  at time  $t$ .
- Let  $J(t, A)$  be the subset of elements of  $A$  that achieve the minimum  $I(t, A)$ .
- Let  $S(i, A) := \sup(t : i \notin J(t, A))$ .
- Define  $e_i$  to be the policy that sends at time  $t$  packet of type  $i$  with probability 1 and does not send packets of other types.

[TABLE 2 about here.]

Algorithm A in Table II strives for equalizing the less populated packets at each point in time: it first increases the CCI of the latest arrived packet, trying to increase it to the minimum CCI which was attained over all the packets existing before the last one arrived (step A3.2). If the minimum is reached (at some threshold  $s$ ), then it next increases the fraction of all packets currently having minimum CCI, seeking now to equalize towards the second smallest CCI, sharing equally the forwarding probability among all such packets. The process is repeated until the next packet arrives: hence, the same procedure is applied over the novel interval. Moreover, it holds the following:

- Theorem 3.3:* Let  $\mathbf{u}^*$  be the policy obtained by Algorithm A when substituting there  $\tau = \infty$ . Then
- (i)  $\mathbf{u}^*$  is optimal for P2.
  - (ii) Consider some finite  $\tau$ . If in addition  $\int_0^\tau X^i(t) dt$  are the same for all  $i$ 's, then  $\mathbf{u}^*$  is optimal for P1.

**Proof:** The proof for (i) is given in the extended version of the paper [16]. That for (ii) is the same as for Theorem 3.1.  $\diamond$

#### IV. BEYOND WORK-CONSERVING POLICIES

We have obtained the structure of the best work-conserving policies, and identified their structure, and identified cases in which these are globally optimal. We next show the limitation of work-conserving policies.

##### A. The case $K=2$

Consider two packets, arriving at the source at  $t_1$  and  $t_2$ , respectively. Consider the policy  $\mu(s)$  where  $0 = t_1 < s \leq t_2$  which transmits packet 1 during  $[t_1, s)$ , does not transmit anything during  $[s, t_2)$  and then transmits packet 2 after  $t_2$ . Let us define  $X(t) = 1 - \exp(-\beta t)$ . Then it holds

$$X_1(t) = \begin{cases} X(t) & 0 \leq t \leq s \\ X(s) & s \leq t \leq \tau \end{cases} ; \quad X_2(t) = \begin{cases} 0 & 0 \leq t \leq t_2 \\ X(t - (t_2 - s)) - X(s) = \\ e^{-\beta s} - e^{-\beta(t - (t_2 - s))} & t_2 \leq t \leq \tau \end{cases}$$

This gives

$$\int_0^\tau X_1(t)dt = \frac{-1 + \beta s + e^{-\beta s}}{\beta} + (\tau - s)(1 - e^{-\beta s}) ; \quad \int_0^\tau X_2(t)dt = \frac{e^{-\beta s}}{\beta}(\beta(\tau - t_2) - 1 + e^{-\beta(\tau - t_2)})$$

*Example 4.1:* Using the above dynamics, we can illustrate the improvement that non work-conserving policies can bring. We took  $\tau = 1$ ,  $t_1 = 0$ ,  $t_2 = 0.8$ . We vary  $s$  between 0 and  $t_2$  and compute the probability of successful delivery for  $\beta = 1, 3, 8$  and 15. The corresponding optimal policies  $u(s)$  are given by the thresholds  $s = 0.242, 0.242, 0.265, 0.425$ . The probability of successful delivery under the threshold policies  $u(s)$  are depicted in Figure 1 as a function of  $s$  which is varied between 0 and  $t_2$ .

[Fig. 1 about here.]

In all these examples, there is no optimal policy among those that are work-conserving.

Solving for the  $t_{eq}$  value, it turns out that a work-conserving policy is optimal for all  $\beta \leq 0.9925$ . Note that under any work-conserving policy,  $\int_0^\tau X_2(t)dt \leq \tau(1 - X(t_2))$  (where  $X(t_2)$  is the same for all work-conserving policies). Now, as  $\lambda$  (hence  $\beta$ ) increases to infinity,  $X(t_2)$  and hence  $X_1(t_2)$  increase to one. Thus  $\int_0^\tau X_2(t)dt$  tends to zero. We conclude that the success delivery probability tends to zero, uniformly under any work-conserving policy.

*Remark 4.1:* The possible improvement in the delivery probability for given  $\tau$  brought by non-WC policies over WC policies relates to the buffer size limitation, indeed a typical constraint which occurs in practice. This is apparent in the case of  $K = 2$ : the dynamics of the number of packets of type 2 is subject to the initial conditions, i.e., number of free relays, found at time  $t_2$  and so does the delivery probability, since forwarding a large number of packets of type 1 makes forwarding of packets of type 2 very slow from  $t_2$  on. In turn,  $P_s$  remains small, impairing the performance of the system. One may think of overcoming this constraint in the system dynamics handling buffers with more flexible policies, e.g., overwriting packets once in the node's buffer. Indeed, packets spreading can keep on going in a work-conserving fashion until the next packet arrives at the source, as the source can use relays with full buffers by forcing them to erase one of their packets so as to make room for the new packet to be carried. Observe that the policies that we would obtain in turn are equivalent in terms of performance, but would come indeed at higher cost in terms of resources, since they would require much higher number of packet transmissions.

### B. Time changes and policy improvement

*Lemma 4.1:* Let  $p < 1$  be some positive constant. For any multi-policy  $\mathbf{u} = (u_1(t), \dots, u_n(t))$  satisfying  $u = \sum_{i=1}^n u_i(t) \leq p$  for all  $t$ , define the policy  $\mathbf{v} = (v_1, \dots, v_n)$  where  $v_i = u_i(t/p)/p$  or equivalently,  $u_i = p v_i(tp)$ ,  $i = 1, \dots, n$ . Define by  $X_i$  the state trajectories under  $\mathbf{u}$ , and let  $\bar{X}_i$  be the state trajectories under  $\mathbf{v}$ . Then  $X(t) = \bar{X}(tp)$ .

**Proof.** We have

$$\frac{d\bar{X}(s)}{ds} = v(s)\beta(1 - \bar{X}(s))$$

where  $v = \sum_{i=1}^n v_i$ . Substituting  $s = tp$  we obtain

$$\frac{d\bar{X}(s)}{dt} = p \frac{d\bar{X}(s)}{ds} = p v(s)\beta(1 - \bar{X}(s)) = u(t)\beta(1 - \bar{X}(s))$$

We conclude that  $X(t) = \bar{X}(tp)$ . Moreover,

$$\frac{d\bar{X}_i(s)}{dt} = p \frac{d\bar{X}_i(s)}{ds} = p v_i(s)\beta(1 - \bar{X}(s)) = u_i(t)\beta(1 - \bar{X}(s))$$

We thus conclude that  $X_i(t) = \bar{X}_i(tp)$  for all  $i$ . ◇

The control  $\mathbf{v}$  in the Lemma above is said to be an *accelerated* version of  $\mathbf{u}$  from time zero with an accelerating factor of  $1/p$ . An acceleration  $\mathbf{v}$  of  $\mathbf{u}$  from a given time  $t'$  is defined similarly as  $v_i(t) = u_i(t)$  for  $t \leq t'$  and  $v_i(t) = u_i(t' + (t - t')/p)/p$  otherwise, for all  $i = 1, \dots, n$ . We now introduce the following policy improvement procedure.

*Definition 4.1:* Consider some policy  $\mathbf{u}$ . and let  $u := \sum_{j=1}^n u_j(t)$ . Assume that  $u \leq p$  over some  $0 < p < 1$  for all  $t$  in some interval  $S = [a, b]$  and that  $\int_b^c u(t)dt > 0$  for some  $c > b$ . Let  $w$  be the policy obtained from  $u$  by

- (i) accelerating it at time  $b$  by a factor of  $1/p$ ,
- (ii) from time  $d := a + p(b - a)$  till time  $c - (1 - p)(b - a)$ , use  $w(t) = u(t + b - d)$ . Then use  $w(t) = 0$  till time  $c$ .

Let  $X(t)$  be the state process under  $u$ , and let  $\bar{X}(t)$  be the state process under  $w$ . Then it is immediate to verify that the following conditions hold:

*Lemma 4.2:* Consider the above policy improvement of  $u$  by  $w$ . Then

- (a)  $\bar{X}_i(t) \geq X_i(t)$  for all  $0 \leq t \leq c$ ,
- (b)  $X_i(c) = \bar{X}_i(c)$  for all  $i$ ,
- (c)  $\int_a^c X_i(t)dt \leq \int_a^c \bar{X}_i(t)dt$ .

### C. General optimal policies

*Theorem 4.1:* Let  $K \geq 2$ . Then an optimal policy for P1, named *piecewise threshold policy*, exists with the following structure:

- (i) There are thresholds,  $s_i \in [t_i, t_{i+1}]$ ,  $i = 1, \dots, K$ . During the intervals  $[s_i, t_{i+1})$  no packets are transmitted.
- (ii) Algorithm A to decide what packet is transmitted at the remaining times; the forwarding probability used there is  $u_t = 1$  for  $t \in [t_i, t_{i+1}]$ ,  $i = 1, \dots, K$ .
- (iii) After time  $t_K$  it is optimal to always transmit a packet. An optimal policy  $u$  satisfies  $u(t) = 1$  for all  $t \geq t_K$ .

**Proof.** (i) Let  $\mathbf{u}$  be an arbitrary policy. Define  $u(t) = \sum_j u_j(t)$ . Assume that it does not satisfy (i) above. Then there exists some  $i = 1, \dots, K - 1$ , such that  $\mathbf{u}(t)$  is not a threshold policy on the interval  $T_i := [t_i, t_{i+1})$ . Hence there is a close interval  $S = [a, b] \subset T_i$  such that for some  $p < 1$ ,  $u(t) \leq p$  for all  $t \in S$  and  $\int_b^{t_{i+1}} u(t)dt > 0$ . Then  $\mathbf{u}$  can be strictly improved according to Lemma 4.2 and hence cannot

be optimal.

(ii) At the remaining times, the policy considered is in fact work-conserving.

(iii) By part (i) the optimal policy has a threshold type on the interval  $[t_K, t_{K+1}]$ . Assume that the threshold  $s$  satisfies  $s < t_{K+1}$ . It is direct to show that by following  $\mathbf{u}$  till time  $s$  and then switching to any policy that satisfies  $u_i(t) > 0$  for all  $i$ ,  $P_s(\tau)$  strictly increases.  $\diamond$

## V. THE CONSTRAINED PROBLEM

Let  $\mathbf{u}$  be any policy that achieves the constraint  $\mathcal{E}(\tau) = \varepsilon x$  as defined in Section II-B. We make the following observation. The constraint involves only  $X(t)$ . It thus depends on the individual  $X_i(t)$ 's only through their sum; the sum  $X(t)$ , in turn, depends on the policies  $u_i$ 's only through their sum  $u = \sum_{i=1}^K u_i$ .

**Work conserving policies.** If a policy is work-conserving and has to meet an energy constraint, then it is such that:  $u = 1$  till some time  $s$  and is then zero.  $s$  is the solution of  $X(s) = z + x$ , i.e.,  $s = -\frac{1}{\beta} \log\left(\frac{1-x-z}{1-z}\right)$ . Algorithm A can be used to generate the optimal policy components  $u_i(t)$ ,  $i = 1, \dots, K$ : in particular, it will perform the same type of equalization performed in the unconstrained case until the bound is reached and it will stop thereafter.

**General policies.** Any policy  $\mathbf{u}$  that is not of the form as described by (i)-(ii) in Theorem 4.1 can be strictly improved by using Lemma 4.2. Thus the optimal policies satisfying a certain energy constraint is piecewise threshold, except that (iii) of Theorem 4.1 needs not to hold.

## VI. ADDING FIXED AMOUNT OF REDUNDANCY WITH OPTIMAL BLOCK-CODES

We now consider adding forward error correction: we add  $H$  redundant packets and consider the new file that now contains  $K + H$  packets. The channel between the source and the destination up to time  $\tau$  can be seen as an erasure channel. For such channel, maximum-distance separable erasure codes exist, such as Reed-Solomon codes, for which it is sufficient to receive any  $K$  packets out of the  $K + H$  to ensure successful decoding of the entire file at the receiver (i.e., retrieval of the  $K$  information packets). We assume that all redundant packets are available (created at the source) once all  $K$  information packets have been received, that is from  $t_K$  onwards. The same results would hold if coded packets are created at separate times  $t_i > t_K$  for  $i = K + 1, \dots, K + H$ . Now we introduce the result that specifies how to optimize work-conserving policies when block codes are adopted.

*Theorem 6.1:* (i) Assume that there exists some policy  $\mathbf{u}$  such that  $\sum_{i=1}^{K+H} u_i(t) = 1$  for all  $t$ , and such that  $Z_i(\tau)$  is the same for all  $i = 1, \dots, K + H$  under  $\mathbf{u}$ . Then  $\mathbf{u}$  is optimal for P1.

(ii) Algorithm A, with  $K + H$  replacing  $K$  and taking  $\tau = \infty$ , produces a policy which is optimal for P2.

**Proof:**

For  $i = 1, \dots, K + H$ , let  $p_i$  be the probability of successful delivery of packet  $i$  by  $\tau$ :  $p_i = \zeta(Z_i) = 1 - \exp(-\lambda \int_0^\tau X_i(t) dt)$ . Let  $E$  be a set of  $|E|$  pairwise different elements of  $\{1, \dots, K + H\}$ . The probability of delivery of the  $K$  packets by time  $\tau$  is given by:

$$P_s(\tau, K, H) = \sum_{Z=K}^{K+H} \sum_{E:|E|=Z} \prod_{i \in E} p_i \prod_{i \notin E} (1 - p_i)$$

Consider any pair of different elements  $i, j \in \{1, \dots, K + H\}$ . Denote  $V = \{1, \dots, K + H\} \setminus \{i, j\}$ . The success probability can be decomposed as

$$\begin{aligned} P_s(\tau, K, H) &= p_i p_j \sum_{\substack{h \subseteq V \\ |h| \geq K-2}} \prod_{r \in h} p_r \prod_{s \in V \setminus h} (1 - p_s) + p_i (1 - p_j) \sum_{\substack{h \subseteq V \\ |h| \geq K-1}} \prod_{r \in h} p_r \prod_{s \in V \setminus h} (1 - p_s) \\ &+ (1 - p_i) p_j \sum_{\substack{h \subseteq V \\ |h| \geq K-1}} \prod_{r \in h} p_r \prod_{s \in V \setminus h} (1 - p_s) + (1 - p_i)(1 - p_j) \sum_{\substack{h \subseteq V \\ |h| \geq K}} \prod_{r \in h} p_r \prod_{s \in V \setminus h} (1 - p_s) \end{aligned} \quad (4)$$

Equation (4) can be rewritten  $P_s(\tau, K, H) = p_i p_j A + p_i (1 - p_j) B + p_j (1 - p_i) C + D$  with obvious meaning of the notation; observe that  $A, B, C, D$  do not depend on  $Z_i$  ( $p_i$ ) and  $Z_j$  ( $p_j$ ). Further rearranging the terms we obtain

$$P_s(\tau, K, H) = (A - B - C + D) p_i p_j + (B - D) p_i + (C - D) p_j + D$$

First, observe that  $B = C$ . Also,  $A \geq B \geq D$ . For any  $i$  and  $j$  in  $\{1, \dots, K + H\}$  we can thus write

$$P_s(\tau, K, H) = g_1 p_i p_j + g_2 (p_i + p_j) + g_3 \quad (5)$$

where  $g_1 = (A - B - C + D)$ ,  $g_2 = (B - D)$  and  $g_3 = D$  are functions only of  $\{p_m, m \neq i, m \neq j\}$ .

Now, fix all  $p_m, m \neq i, j$ : optimizing the work-conserving policy with respect to  $Z_i$  and  $Z_j$  means finding  $Z_i$  and  $Z_j$  minimizing (5), with  $Z_i + Z_j = C = Z_{total} - \sum_{r \neq i, j} Z_r$ . Again,  $C$  does not depend on  $Z_i, Z_j$ .

Thus we have  $p_j = 1 - \frac{e^{-\lambda C}}{1 - p_i}$ . Thus, it follows that  $P_s(\tau, K, H) = f_C(p_i)$ , where

$$f_C(p_i) = g_1 p_i \left(1 - \frac{e^{-\lambda C}}{1 - p_i}\right) + g_2 \left(p_i + \left(1 - \frac{e^{-\lambda C}}{1 - p_i}\right)\right) + g_3$$

and it follows

$$\frac{df_C(p_i)}{dp_i} = ((1 - p_i)^2 - e^{-\lambda C}) \frac{g_1 + g_2}{(1 - p_i)^2},$$

As  $g_1 + g_2 \geq 0$ , the maximum of  $f_C(p_i)$  is attained for  $p_i = 1 - e^{-\lambda C/2}$ . Hence, it is optimal to equalize pairwise  $Z_i$  and  $Z_j$ : since this holds for any  $i$  and  $j$ . Moreover,  $f_C(\cdot)$  is increasing with  $C$ , and a work-conserving policy achieves the highest feasible  $Z_{total}$  by construction. If all  $Z_i$  are equal, then  $C = 2Z_{total}/K$  is also maximum for all pairs  $\{i, j\}$ . This implies (i). Item (ii) can be proven in the same way as in Theorem 3.3.  $\diamond$

Also, the other results that we had for the case of no redundancy can be obtained here as well (those for P1, CP1 and CP2).

## VII. RATELESS CODES

In this section, we want to identify the possible rateless codes for the settings described in Section II, and quantify the gains brought by coding. Rateless erasure codes are a class of erasure codes with the property that a potentially limitless sequence of coded packets can be generated from a given set of information packets; information packets, in turn, can be recovered from any subset of the coded packets of size equal to or only slightly larger than  $K$  (the amount of additional needed packets for decoding is named ‘‘overhead’’). As in the previous section, we assume that redundant packets are created only after  $t_K$ , i.e., when all information packets are available. The case when coding is started before receiving all information packets is postponed to the next section. Since coded packets are generated after all information packets have been sent out, the code must be *systematic* because information packets are part of the coded packets.

Amongst rateless codes, LT codes [17] and Raptor codes [18] are near to optimal in the sense that the overhead can be arbitrarily small with some parameters. The coding matrix of each of them has a specific structure in order to reduce encoding and decoding complexity. Only Raptor codes exist in a systematic version. Network codes [19] are more general rateless codes as generating coded packets relies on random linear combinations (RLCs) of information packets, without any (sparsity) constraint for the matrix of the code. Their overhead can be considered as 0 for high enough field order. That is why in this section we provide the analysis of the optimal control for network codes. But, it is straightforward to extend these results to systematic Raptor codes.

### A. Rateless coding after $t_K$

After  $t_K$ , at each transmission opportunity, the source sends a redundant packet (a RLC of all information packets) with probability  $u$ . Indeed, from  $t_K$ , any sent random linear combination carries the same amount of information of each information packet, and hence from that time, the policy is not function of a specific packet anymore, whereby  $u$  instead of  $\mathbf{u}$ . In each sent packet, a header is added to describe what are the coefficients, chosen uniformly at random, of each information packet. The decoding of the  $K$  information packets is possible at the destination if and only if the matrix made of the headers of received packets has rank  $K$ . Note that, in our case, the coding is performed only by the source since the relay nodes cannot store more than one packet.

*Theorem 7.1:* Let us consider the above rateless coding scheme for coding after  $t_K$ .

- (i) Assume that there exists some policy  $\mathbf{u}$  such that  $\sum_{i=1}^{K-1} u_i(t) = 1$  for all  $t$ , and such that  $Z_i$  is the same for all  $i = 1, \dots, K-1$  under  $\mathbf{u}$ . Then  $\mathbf{u}$  is optimal for P1.
- (ii) Algorithm B produces a policy which is optimal for P2.

[TABLE 3 about here.]

**Proof:** Let  $E$  be a set made of pairwise different elements from  $\{1, \dots, K-1\}$ . Let us take the notation of proof of Theorem 6.1. We have

$$P_s(\tau, K) = \sum_{E \subset \{1, \dots, K-1\}} \left( \prod_{i \in E} p_i \prod_{i \in \{1, \dots, K-1\} \setminus E} (1 - p_i) \right) Q(|E|)$$

where  $Q(|E|)$  denotes the probability that the received coded packets, added to the  $|E|$  received information packets, form a rank  $K$  matrix. Let  $P_m$  be the probability that exactly  $m$  coded packets are received at the destination by time  $\tau$ . Let consider the probability that, given that  $m \geq K - e$  coded packets have been received, these packets form a rank  $K$  matrix with the received  $e$  information packets. Then  $Q(E)$  is expressed as:  $Q(e) = \sum_{m=K-e}^{\infty} P_m \prod_{r=0}^{K-e-1} \left(1 - \frac{1}{q^{m-r}}\right)$ . It can be seen that  $Q(|E|)$  depends on  $u_i(t)$ , for  $i = 1, \dots, K-1$ , only through the sum  $\sum_{i=1}^{K-1} u_i(t)$ , which is 1 (resp. 0) for  $t < t_K$  (resp.  $t \geq t_K$ ). Let  $V = \{1, \dots, K-1\}$  and defined  $Q(\geq a) = \sum_{e=a}^{\infty} Q(e)$  for the sake of notation. We have

$$\begin{aligned} P_s(\tau, K) &= p_i p_j \sum_{\substack{h \subset V \\ |h| \geq 0}} Q(\leq K-2-h) \prod_{r \in h} p_r \prod_{s \in V \setminus h} (1 - p_s) + p_i(1 - p_j) \sum_{\substack{h \subset V \\ |h| \geq K-1}} Q(\leq K-1-h) \prod_{r \in h} p_r \prod_{s \in V \setminus h} (1 - p_s) \\ &+ (1 - p_i) p_j \sum_{\substack{h \subset V \\ |h| \geq K-1}} Q(\leq K-1-h) \prod_{r \in h} p_r \prod_{s \in V \setminus h} (1 - p_s) + (1 - p_i)(1 - p_j) \sum_{\substack{h \subset V \\ |h| \geq K}} Q(\leq K-h) \prod_{r \in h} p_r \prod_{s \in V \setminus h} (1 - p_s) \end{aligned}$$



Therefore, the definition of constants A, B, C and D are different than in proof of Theorem 6.1, but their properties remain the same. Thus, in the same way, we can derive the desired result.  $\diamond$

### B. Rateless coding before $t_K$

We now consider the case where after receiving packet  $i$  and before receiving packet  $i+1$  at the source, we allow to code over the available information packets and to send resulting coded packets between  $t_i$  and  $t_{i+1}$ . LT codes and Raptor codes require that all the information packets are available at the source before generating coded packets. Owing to their fully random structure, network codes do not have this constraint, and allow to generate coded packets online, along the reception of packets at the source. We present how to use network codes in such a setting. The objective is the successful delivery of the entire file (the  $K$  information packets) by time  $\tau^1$ . Information packets are not sent anymore, only coded packets are sent instead.

*Theorem 7.2:* (i) Given any forwarding policy  $u(t)$ , it is optimal for P1 and P2 to send coded packets resulting from random linear combinations of all the information packets available at the time of the transmission opportunity with probability  $u(t)$ .

(ii) For any policy  $u(t)$ , the probability of successful delivery of the entire file is given by

$$P_s(\tau) = \sum_{j=0}^{K-1} \sum_{k_1 > \dots > k_j} \sum_{l_0=K-k_1}^K \dots \sum_{l_i=K-k_{i+1}-L_{i-1}}^{K-L_{i-1}} \dots \sum_{l_j=K-L_{j-1}}^{k_j} \sum_{m_0=l_0}^{\infty} \dots \sum_{m_j=l_j}^{\infty} \prod_{i=0}^j f(l_0, \dots, l_i, m_i),$$

with  $f(l_0, \dots, l_i, m_i) = P(m_k; u(t)) \binom{m_i}{l_i} \prod_{r=0}^{l_i-1} \left(1 - \frac{1}{q^{K-L_{i-1}-r}}\right)$ , where

$$P(m_k; u(t)) = \exp(-\Lambda_k) \frac{\Lambda_k^{m_k}}{m_k!}, \quad \Lambda_k = \lambda \int_0^\tau Y_k(t) dt,$$

$$Y_k(t) = (t \geq t_{k+1}) \lambda \int_{t_k}^{\min(t, t_{k+1})} u(v) \exp\left(-\lambda \int_0^v u(s) ds\right) dv.$$

**Proof:** The proof is based on finding the condition for having full-rank decoding matrix, given that the packets of each class arrive according to a Poisson process. For all  $k = 1, \dots, K$ , let  $E(k)$  be  $E(k) = \{1, \dots, k\}$ . For short, we say that a coded packet is “a packet over  $E(k)$ ” if the coefficients of the first  $k$  information packets are chosen uniformly at random in  $\mathbb{F}_q$ , while the others are zero. We have the following definitions:

- The received packets are over  $E(k_i)$ , with  $K = k_0 > k_1 > k_2 > \dots > k_j \geq 1$ .

<sup>1</sup>We do not have constraints on making available at the destination a part of the  $K$  packets in case the entire file cannot be delivered.

- $j$  is such that  $0 \leq j < K$ , and denotes the number of pairwise different  $k_i \neq K$ ,  $i = 0, \dots, j$ . We set  $k_{j+1} = 0$ .
- $m_i$ ,  $i = 0, \dots, j$  is the number of received packets over  $E(k_i)$ .
- $S_i$  is the sub-matrix made of all the received packets over  $E(k_i)$ .

In order to recover the  $K$  information packets, the matrix which has to be full-rank is composed of the  $S_i$ , for  $i = 0, \dots, j$ . Let  $\text{rrf}(M)$  denote the reduced row form of any matrix  $M$ , i.e., the matrix resulting from Gaussian elimination on  $M$ , without column permutation.

[Fig. 2 about here.]

Let us consider the process described in Figure 2. At each step  $i$ ,  $i = 0, \dots, j$ ,  $l_i$  is defined as the number of non-zero rows of  $\text{rrf}(P)$ . Therefore, for the coding matrix to have rank  $K$ , it is necessary and sufficient to have:  $l_0 \geq K - k_1$ ,  $l_1 \geq k_1 - k_2 - (l_0 - (K - k_1))$ ,  $\dots$ ,  $l_i \geq K - k_{i+1} - L_{i-1}$ ,  $l_j \geq K - L_{j+1}$ . Therefore we can express  $P_s(\tau)$ :

$$P_s(\tau) = \sum_{j=0}^{K-1} \sum_{k_1 > \dots > k_j} \sum_{l_0=K-k_1}^K \dots \sum_{l_i=K-k_{i+1}-L_{i-1}}^{K-L_{i-1}} \dots \sum_{l_j=K-L_{j+1}}^{k_j} \sum_{m_0=l_0}^{\infty} \dots \sum_{m_j=l_j}^{\infty} \prod_{i=0}^j f(l_0, \dots, l_i, m_i),$$

where  $f(l_0, \dots, l_i, m_i)$  is a joint probability:  $f(l_0, \dots, l_i, m_i) = P(l_i, m_i | l_0, \dots, l_{i-1}) = P(l_i | m_i, l_0, \dots, l_{i-1})P(m_i)$ .

We have  $P(l_i | m_i, l_0, \dots, l_{i-1}) = \binom{m_i}{l_i} \prod_{r=0}^{l_i-1} \left(1 - \frac{1}{q^{K-L_{i-1}-r}}\right)$ .  $P(m_i)$  is the probability that  $m_i$  coded packets over  $E(k_i)$  have reached the destination (by time  $\tau$ ). This probability is therefore dependent on  $u_i(t)$ , for  $i = 0, \dots, j$ .

Now we are able to analyze how must the  $k_i$  be chosen, for  $i = 0, \dots, j$ , so as to maximize  $P_s(\tau)$ , when the other system parameters are fixed. For every  $i = 0, \dots, j$ , we can see from the above expression of  $P_s(\tau)$  that it is maximized if  $l_i$  needs to be as less as possible, i.e., when  $k_{i+1}$  and  $L_{i-1}$  are maximized. Note that once  $k_i$  is set,  $l_i$  is only dependent on  $u_i(t)$  and the mobility process (transmission opportunities). Thus, for given  $i$ , the probability to receive the required  $l_i$  is maximized for maximized  $k_{i+1}$ . Thus it is optimal to send packets coded over all the available information packets. Therefore  $u_i(t)$  is  $u(t)$  for  $i$  such that  $k_i$  is maximum at time  $t$ , 0 otherwise. This proves part (i) of the theorem.

Owing to the mobility process and the buffer size limited to one packet, the arrival process of the packets over  $E(k)$  at the destination is a non-homogeneous Poisson process of parameter  $\Lambda_k(t)$ . Let  $\Lambda_k = \Lambda_k(\tau)$ . We have:  $P(m_k; u(t)) = \exp(-\Lambda_k) \frac{\Lambda_k^{m_k}}{m_k!}$ , where  $\Lambda_k = \lambda \int_0^\tau Y_k(t) dt$ . Then the expression of  $Y_k(t)$  is derived thanks to fluid approximation (see the extended version of the paper [16]).  $\diamond$

## VIII. NUMERICAL ANALYSIS

### A. Numerical analysis of work-conserving policies

Comparing the successful delivery probabilities for the different coding schemes, it can be seen that coding allows simplification of the scheduling policy, with or without energy constraint. Figure 3 is an example of numerical comparison between the four coding schemes for work-conserving policies designed with Algo. A. We can conclude that (i) as soon as coding is performed, it saves the source maintaining states  $Z_i(t)$ , and (ii) the sooner coding is performed at the source, the better.

[Fig. 3 about here.]

### B. Numerical optimization of threshold policies

Previous sections have presented analytic results about optimization of work-conserving policies. However, Theorem 4.1 states that a work-conserving policy may not always attain the maximum of the delivery probability. The maximum is always attained by a so called threshold policy. The threshold policy turns into a WC policy when each thresholds  $s_i$  is equal to  $t_{i+1}$ , for  $i = 1, \dots, K - 1$ .

Figure 4 shows the success probability of threshold policies, under various coding schemes and for  $K = 3$ , as well as the variation of best  $s_1$  against  $\tau$ . For each value of  $\tau$ , the two-dimensional optimization of the parameters  $(s_1, s_2)$  pertains to the class of nonlinear optimization problems. Many general algorithms for solving such problems have been developed. We experimented with an algorithm called Differential Evolution (DE) [20]. DE is a robust optimizer for multivariate functions. We do not describe DE here, but only say that this algorithm is in part a hill climbing algorithm and in part a genetic algorithm. This optimization does not have to be performed at the source node, but is rather performed offline, and the resulting optimal parameters for each interesting value of  $\tau$  are stored in memory at the source node, so as to be used as needed.

We do not comment  $s_2$  as it is equal to  $t_3$  for all coding schemes and all values of  $\tau$ . We can interpret the variations of  $s_1$  thanks to the comments made after Theorem 3.1. Let us first give some general comments holding for all coding schemes. We can see that  $s_1$  is relatively high when  $\tau$  is close to  $t_3$ . Indeed, in such case, the limiting parameter for  $Z_3(\tau)$  (and  $Z_2(\tau)$ ) is  $\tau - t_3$  (or  $\tau - t_2$ ) instead of  $X_1(s_1)$ , and hence it is better to spread as many packet 1 as possible. Then  $s_1$  starts to decrease: as  $\tau$  increases,  $\tau - t_3$  stops progressively to be the limiting factor for  $Z_3(\tau)$ , and that becomes to be  $X_1(s_1)$  which has

hence to be limited. When  $\tau$  increases even more,  $s_1$  can get higher again because copies of packet 3 have enough time to spread enough, even though they spread slower because of higher  $X_1(s_1)$  from  $t = t_3$ .

Now regarding the specificities of each coding schemes, we can notice that the sooner coding is performed, the lower  $s_1$  for all values of  $\tau$ . As proven in the above section, it is better to propagate coded packets resulting from combination over as many as possible original packets. Therefore, when coding is used, coded packets sent out later must be more useful to recover any of the previously received packets. That can be a reason for  $s_1$  to be that low for coding schemes.

TO ADD: As tau increases and we have more choices for  $s_1$ , it can be better to not spread to many packets with only packet 1 information, but wait a bit so as to send packets with packet 1 and packet 2 information, and those coded packets will be more likely to be useful to the destination.

[Fig. 4 about here.]

## IX. CONCLUSIONS

In this paper we have addressed the problem of optimal transmission policies in DTN with two-hop routing under memory and energy constraints. We tackled the fundamental scheduling problem that arises when several packets that compose the same file are available at the source at different time instants. The problem is then how to optimally schedule and control the forwarding of such packets in order to maximize the delivery probability of the entire file to the destination. We solved this problem both for work-conserving and non work-conserving policies, deriving in particular the structure of the general optimal forwarding control that applies at the source node. Furthermore, we extended the theory to the case of fixed rate systematic erasure codes and random linear codes. Our model includes both the case when coding is performed after all the packets are available at the source, and also the important case of random linear codes, that allows for dynamic runtime coding of packets as soon as they become available at the source.

## REFERENCES

- [1] E. Altman and F. De Pellegrini, "Forward correction and fountain codes in delay tolerant networks," in *Proc. of Infocom*, April 2009.
- [2] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proc. of SIGCOMM workshop on Delay-tolerant networking (WDTN)*. Philadelphia, Pennsylvania, USA: ACM, 2005.
- [3] E. Altman, T. Başar, and F. De Pellegrini, "Optimal monotone forwarding policies in delay tolerant mobile ad-hoc networks," in *Proc. of ACM/ICST Inter-Perf.* Athens, Greece: ACM, October 24 2008.

- [4] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 109–120, 2005.
- [5] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *Proc. of SIGCOMM workshop on Delay-tolerant networking (WDTN)*. Philadelphia, Pennsylvania, USA: ACM, August 26 2005, pp. 229–236.
- [6] C. Fragouli, J.-Y. L. Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 63–68, 2006.
- [7] Y. Lin, B. Liang, and B. Li, "Performance modeling of network coding in epidemic routing," in *Proc. of MobiSys workshop on Mobile opportunistic networking (MobiOpp)*. San Juan, Puerto Rico: ACM, June 11 2007, pp. 67–74.
- [8] Y. Lin, B. Li, and B. Liang, "Efficient network coded data transmissions in disruption tolerant networks," in *IEEE Conference on Computer Comm. (INFOCOM)*, Phoenix, AZ, Apr. 2008.
- [9] J. Widmer and J.-Y. L. Boudec, "Network coding for efficient communication in extreme networks," in *Proc. of the ACM SIGCOMM workshop on Delay-tolerant networking (WDTN)*, Philadelphia, Pennsylvania, USA, August 26 2005, pp. 284–291.
- [10] R. Groenevelt and P. Nain, "Message delay in MANETs," in *Proc. of SIGMETRICS*. Banff, Canada: ACM, June 6 2005, pp. 412–413, see also R. Groenevelt, *Stochastic Models for Mobile Ad Hoc Networks*. PhD thesis, University of Nice-Sophia Antipolis, April 2005.
- [11] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Elsevier Computer Networks*, vol. 51, pp. 2867–2891, July 2007.
- [12] T. G. Kurtz, "Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes," *Journal of Applied Probability*, vol. 7, no. 1, pp. 49–58, 1970.
- [13] M. Benaïm and J.-Y. Le Boudec, "A class of mean field interaction models for computer and communication systems," *Performance Evaluation*, vol. 65, no. 11-12, pp. 823–838, 2008. [Online]. Available: <http://infoscience.epfl.ch/record/121369/files/pe-mf-tr.pdf>
- [14] E. Altman, G. Neglia, F. De Pellegrini, and D. Miorandi, "Decentralized stochastic control of delay tolerant networks," in *Proc. of Infocom*, April 2009.
- [15] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications*. Academic Press, 1979.
- [16] E. Altman, F. De Pellegrini, and L. Sassatelli, "Dynamic control of coding for general packet arrivals in dtn," 2012. [Online]. Available: [http://www.i3s.unice.fr/~sassatelli/APS12\\_long.pdf](http://www.i3s.unice.fr/~sassatelli/APS12_long.pdf)
- [17] M. Luby, "LT Codes," in *Proc. 43rd IEEE Symp. Foundations of Computer Sciences*, Vancouver BC, Canada, November 2002.
- [18] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.
- [19] D. S. Lun, M. Médard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, September 2004, pp. 20–29.
- [20] K. Price and R. Storn, "Differential Evolution: A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces," *J. Global Optimiz.*, vol. 11, pp. 341–359, 1997.

## LIST OF FIGURES

1	$K = 2, \beta = 1, 3, 8, 15$ . (a) Success probability under non work-conserving policy $u(s)$ as a function of $s$ ; top curve corresponds to largest value of $\beta$ ; second top corresponds to second largest $\beta$ etc. (this order changes only at $s$ very close to 0.5). (b) The evolution of $X(t)$ as a function of $t$ under the best work-conserving policy. The curves are ordered according to $\beta$ with the top curve corresponding to the largest $\beta$ etc. . . . . .	22
2	The decoding process formatted so as to express the successful decoding condition. At each step $i, i = 0, \dots, j, l_i$ is defined as the number of non-zero rows of $\text{rrf}(P)$ . . . . .	23
3	$P_s(\tau)$ for WC policies based on Algo. A, under various coding schemes. No energy constraint. Parameters are $N = 100, \lambda = 2 \cdot 10^{-5}$ , (a) $K = 10, \mathbf{t} = (119, 1299, 1621, 1656, 3112, 3371, 4693, 5285, 5688, 7999)$ , (b) $K = 4, \mathbf{t} = (1000, 5000, 7000, 20000)$ . . . . .	24
4	Parameters are $N = 100, \lambda = 2 \cdot 10^{-5}, K = 3, \mathbf{t} = (1, 7000, 10000)$ , no energy constraint.(b) $P_s(\tau)$ for an optimized threshold policy. (c) Value of threshold $s_1(\tau)$ . . . . .	25

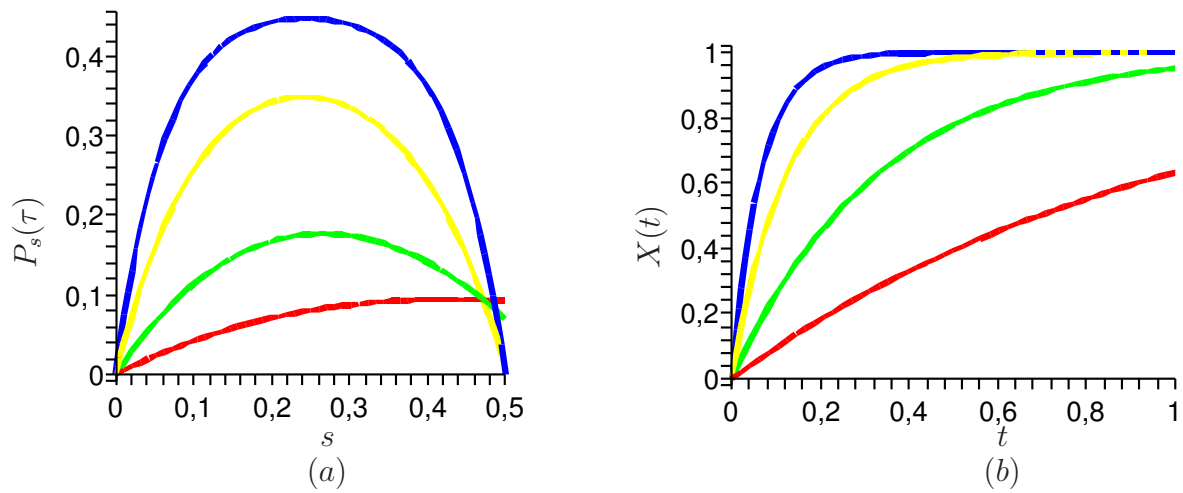


Fig. 1.  $K = 2$ ,  $\beta = 1, 3, 8, 15$ . (a) Success probability under non work-conserving policy  $\mathbf{u}(s)$  as a function of  $s$ ; top curve corresponds to largest value of  $\beta$ ; second top corresponds to second largest  $\beta$  etc. (this order changes only at  $s$  very close to 0.5). (b) The evolution of  $X(t)$  as a function of  $t$  under the best work-conserving policy. The curves are ordered according to  $\beta$  with the top curve corresponding to the largest  $\beta$  etc.

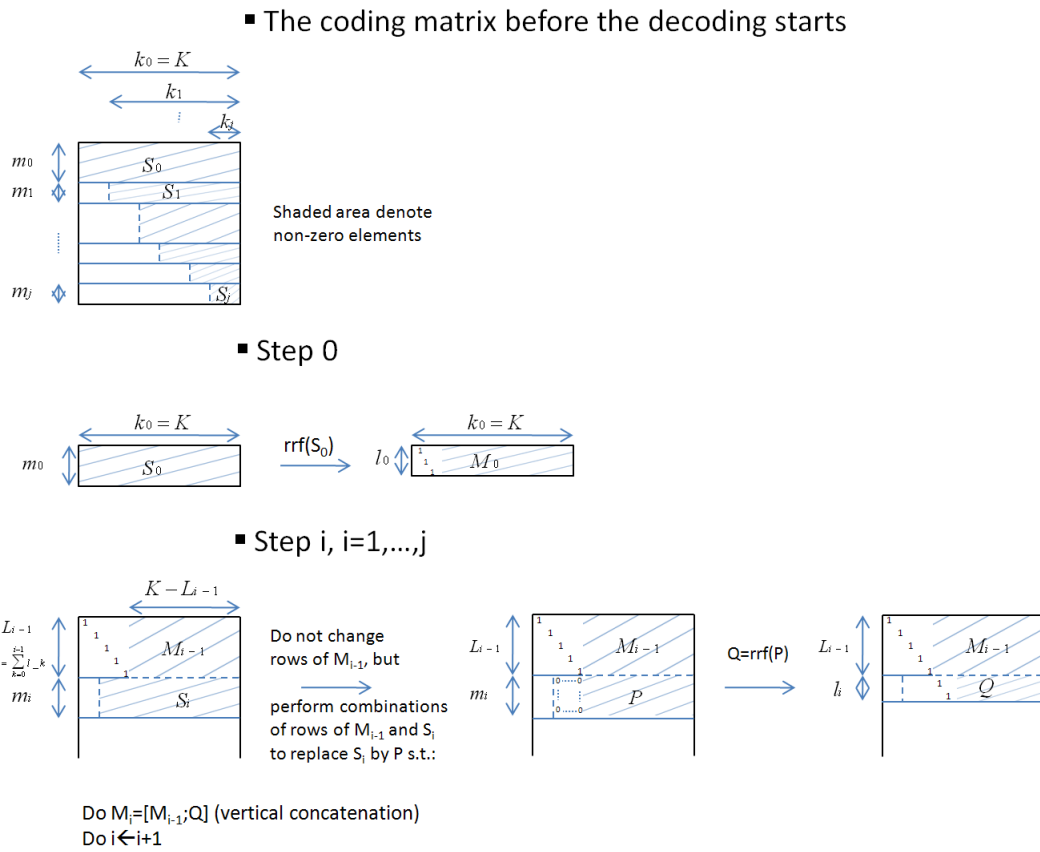


Fig. 2. The decoding process formatted so as to express the successful decoding condition. At each step  $i$ ,  $i = 0, \dots, j$ ,  $l_i$  is defined as the number of non-zero rows of  $\text{rrf}(P)$ .



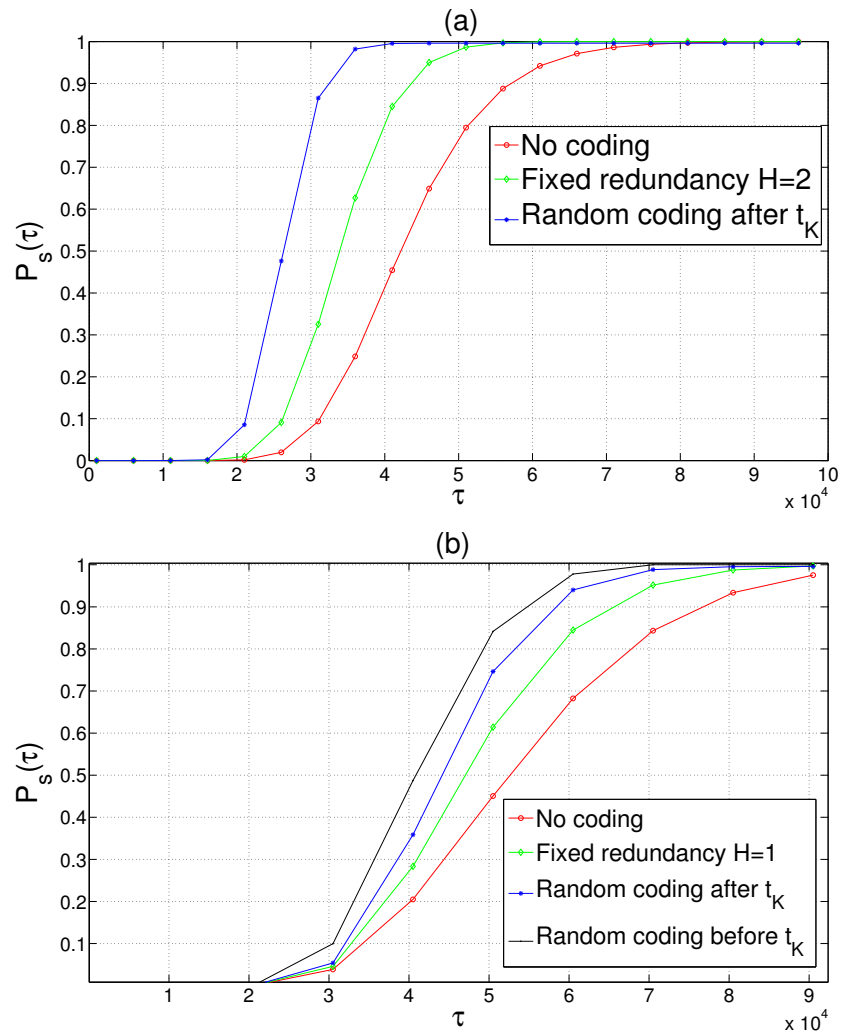


Fig. 3.  $P_s(\tau)$  for WC policies based on Algo. A, under various coding schemes. No energy constraint. Parameters are  $N = 100$ ,  $\lambda = 2.10^{-5}$ , (a)  $K = 10$ ,  $\mathbf{t} = (119, 1299, 1621, 1656, 3112, 3371, 4693, 5285, 5688, 7942)$ . (b)  $K = 4$ ,  $\mathbf{t} = (1000, 5000, 7000, 20000)$ .

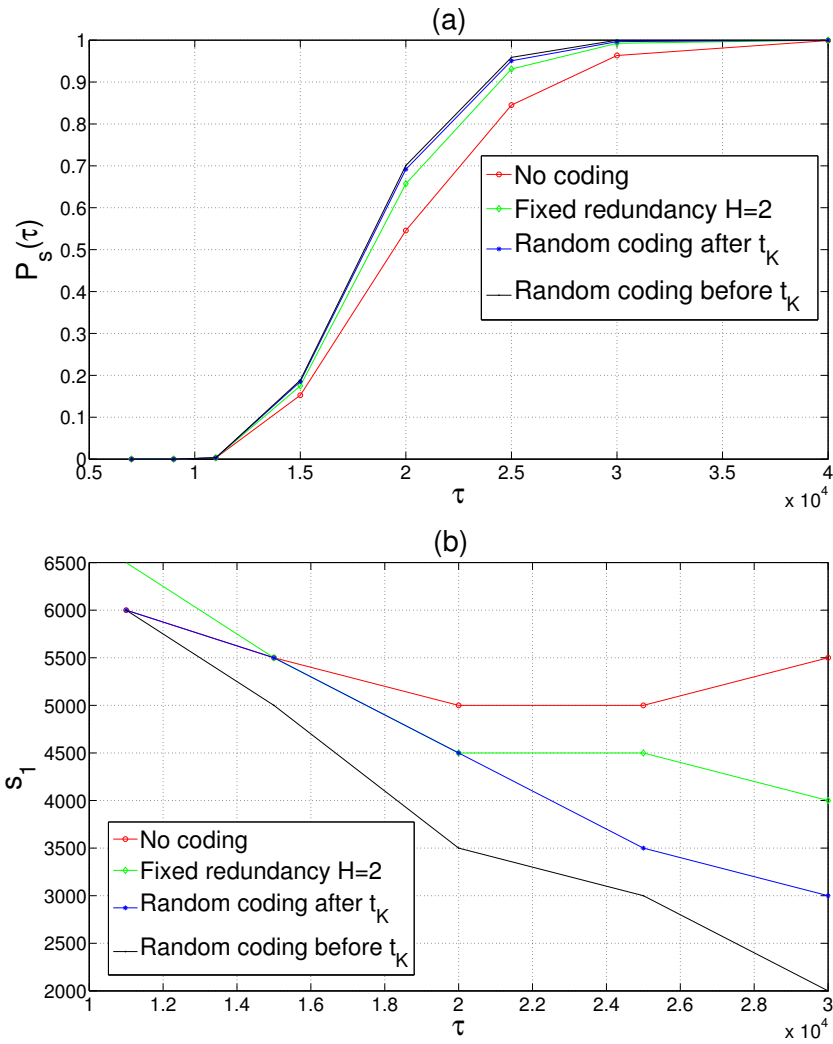


Fig. 4. Parameters are  $N = 100$ ,  $\lambda = 2.10^{-5}$ ,  $K = 3$ ,  $\mathbf{t} = (1, 7000, 10000)$ , no energy constraint. (b)  $P_s(\tau)$  for an optimized threshold policy. (c) Value of threshold  $s_1(\tau)$ .

## LIST OF TABLES

I	Main notation used throughout the paper . . . . .	27
II	Algorithm A . . . . .	28
III	Algorithm B . . . . .	29

TABLE I  
MAIN NOTATION USED THROUGHOUT THE PAPER

<i>Symbol</i>	<i>Meaning</i>
$\bar{N}$	number of nodes (excluding the destination)
$K$	number of packets composing the file
$H$	number of redundant packets
$\lambda$	inter-meeting intensity
$\tau$	timeout value
$X_i(t)$	fraction of nodes (excluding the destination) having packet $i$ at time $t$
$X(t)$	summation $\sum_i X_i(t)$
$\hat{X}_i, \hat{X}$	corresponding sample paths
$z$	$:=X(0)$ will be taken 0 unless otherwise stated.
$u_i(t)$	forwarding policy for packet $i$ ; $\mathbf{u} = (u_1, u_2, \dots, u_K)$
$u$	sum of the $u_i$ s
$Z_i(t), Z_i$	$Z_i(t) = \int_0^T X_i(u) du$ , $Z_i = Z_i(\tau)$ , $\mathbf{Z}(t) = (Z_1(t), Z_2(t), \dots)$ , $\mathbf{Z} = \mathbf{Z}(\tau)$ , $Z = \sum Z_i$
$D_i(\tau)$	probability of successful delivery of packet $i$ by time $\tau$
$P_s(\tau)$	probability of successful delivery of the file by time $\tau$ ; $P_s(\tau, K, H)$ is used to stress the dependence on $K$ and $H$
$\mathbb{R}_+$	nonnegative real numbers

TABLE II  
ALGORITHM A

- A1 Use  $\mathbf{p}_t = e_1$  at time  $t \in [t_1, t_2)$ .
- A2 Use  $\mathbf{p}_t = e_2$  from time  $t_2$  till  $s(1, 2) = \min(S(2, \{1, 2\}), t_3)$ . If  $s(1, 2) < t_3$  then switch to  $\mathbf{p}_t = \frac{1}{2}(e_1 + e_2)$  till time  $t_3$ .
- A3 Define  $t_{K+1} = \tau$ . Repeat the following for  $i = 3, \dots, K$ :
- A3.1 Set  $j = i$ . Set  $s(i, j) = t_i$
- A3.2 Use  $\mathbf{p}_t = \frac{1}{i+1-j} \sum_{k=j}^i e_k$  from time  $s(i, j)$  till  $s(i, j - 1) := \min(S(j, \{1, 2, \dots, i\}), t_{i+1})$ . If  $j = 1$  then end.
- A3.3 If  $s(i, j - 1) < t_{i+1}$  then take  $j = \min(j : j \in J(t, \{1, \dots, i\}))$  and go to step [A3.2].

TABLE III  
ALGORITHM B

- C1 Use  $\mathbf{p}_t = e_1$  at time  $t \in [t_1, t_2)$ .
- C2 Use  $\mathbf{p}_t = e_2$  from time  $t_2$  till  $s(1, 2) = \min(S(2, \{1, 2\}), t_3)$ . If  $s(1, 2) < t_3$  then switch to  $\mathbf{p}_t = \frac{1}{2}(e_1 + e_2)$  till time  $t_3$ .
- C3 Repeat the following for  $i = 3, \dots, K - 1$ :
- C3.1 Set  $j = i$ . Set  $s(i, j) = t_i$
- C3.2 Use  $\mathbf{p}_t = \frac{1}{i+1-j} \sum_{k=j}^i e_k$  from time  $s(i, j)$  till  $s(i, j - 1) := \min(S(j, \{1, 2, \dots, i\}), t_{i+1})$ . If  $j = 1$  then end.
- C3.3 If  $s(i, j - 1) < t_{i+1}$  then take  $j = \min(j : j \in J(t, \{1, \dots, i\}))$  and go to step [C3.2].
- C4 From  $t = t_K$  to  $t = \tau$ , use all transmission opportunities to send a random linear combination of information packets, with coefficients picked uniformly at random in  $\mathbb{F}_q$ .