

Controlled Flooding of Fountain Codes

Waqas bin Abbas, Paolo Casari, *Senior Member, IEEE*, Michele Zorzi, *Fellow, IEEE*

Abstract—We consider a multihop network where a source node must reliably deliver a set of data packets to a given destination node. To do so, the source applies a fountain code and floods the encoded packets through the network, until they reach their destination or are lost in the process. We model the probability that the destination can recover the original transmissions from the received coded packets as a function of the network topology and of the code redundancy, and show that our analytical results predict the outcome of simulations very well. These results are employed to design distributed forwarding policies that achieve a good tradeoff between the success probability and the total number of transmissions required to advance a packet towards the destination. We finally develop in detail the case where intermediate relays can inject additional redundancy in the network, provided that they have successfully decoded the source packets.

Index Terms—Fountain codes; flooding; restricted flooding; heuristic policies; analysis; simulation.

I. INTRODUCTION

FOUNTAIN codes [1], [2] have been introduced to achieve reliable communications in generic networks, where the packet loss probability may not be known a priori. The latter condition makes fountain codes particularly well suited to wireless networks [3], [4], where the ability to dynamically adjust the amount of redundancy offers a means to compensate for link outage events and communication failures [5].

In multihop wireless networks, and especially in those conditions where a high degree of reliability is sought [6], fountain-coded packets may be coupled with some form of redundant network-layer transmission pattern, e.g., flooding [7]. Flooded fountain codes introduce a double layer of redundancy: on one hand, each coded packet is generated by encoding over multiple source packets, and inherently carries information about all of them. As a result, each coded packet is useful not just for an intended relay, but rather it helps all nodes downstream increase their probability to correctly invert the fountain code. Furthermore, each coded source packet is retransmitted by a number of relays, i.e., those that participate in the flooding process. This double layer of redundancy substantially increases the opportunities for each relay between the source and the destination to correctly receive the packets,

and thus be able to retransmit them. However, many replicas of each packet are circulated, possibly leading to a waste of resources.

Fountain codes have been widely considered in the literature as a flexible error control technique in both wired and wireless networks experiencing erasure channels. For example, the SYNAPSE++ protocol [3] employs a fountain code to reliably disseminate firmware over a wireless sensor network. Fountain codes have also been shown to improve the performance of unicast communications, provided that early acknowledgment packets are sent in order to prevent the generation of useless redundancy [8]. In [6], the introduction of fountain codes in vehicular networks leads to a significant performance improvement compared to a simpler approach where data packets are periodically retransmitted. Delay-Tolerant Networks (DTNs) [9] and distributed storage systems [10] also benefit from the distribution of redundancy that comes from the use of fountain codes. Therefore, it is important to design a procedure to disseminate fountain-coded data through a network in an effective yet controlled method. Ideally, such method should exploit the double redundancy offered by coding and flooding in order to deliver packets successfully, while at the same time avoiding excessive packet replication and the resulting energy wastage.

In our recent work [11], we analyzed the interplay between fountain codes and flooding, by considering both a full flooding and a restricted flooding policy. Given the amount of redundancy generated at the source, the number of hops, and the number of relays that interact at each hop, we derive the average number of transmissions required to achieve a prescribed probability of success at the destination. The latter is employed to analyze the interplay between the number of transmissions and the code redundancy.

In this paper, we extend the work in [11] in several respects. First, while we maintain the analysis simple and tractable, we employ a more realistic physical layer model including path loss and fading in our simulation results. We extend our analytical results to forecast the behavior of the network in different topologies and under different deployment assumptions. Our forwarding schemes are evaluated based both on a simple error model and on a more realistic block fading model, which validate our analytical results. Second, we propose a family of simple policies that implement restricted flooding distributedly, based only on the local decisions of each node. We also formulate an adaptive policy that approaches the behavior of full flooding, but with a significantly lower retransmission overhead. Both policies combine forwarding with practical medium access control (MAC)-level considerations that help reduce the complexity and cost of packet forwarding.

The remainder of this paper is organized as follows. In the next section we survey the related literature. In Section III

Manuscript received: May 3, 2016; revised: October 30, 2016, and March 30, 2017; accepted: April 20, 2017.

The associate editor coordinating the review of this manuscript and approving it for publication was Ozan Koyluoglu.

W. bin Abbas (corresponding author, email: waqas.abbas@nu.edu.pk) is with the National University of Computer and Emerging Sciences (NUCES), Islamabad, Pakistan. P. Casari is with the IMDEA Networks Institute, Madrid, Spain. M. Zorzi is with the Department of Information Engineering, University of Padova, Italy. This work was mainly carried out when W. bin Abbas was with the Department of Information Engineering University of Padova, Italy.

Part of this work was presented at the IEEE CAMAD 2014 conference, Athens, Greece [11].

Digital Object Identifier: XX.XXXX/XXX...

we introduce our network model: we analytically characterize the probability of success and the number of transmissions taking place in the network under different packet forwarding schemes, and introduce practical and distributed forwarding policies; in Section IV, we evaluate the performance of these policies and compare them to the results obtained from the analytical model; in Section V we develop in detail one among several possible extensions to our work. Finally, Section VI concludes the paper.

II. RELATED WORK

Several works in the literature have applied fountain codes to scenarios of interest, mostly considering end-to-end communications throughout a network. In [5], an analytical model is provided for the throughput performance of a fountain code transmitted through either a conventional or a cooperative multihop network, both in the delay tolerant and in the delay-constrained case, assuming a Nakagami fading model. The work in [12] derives an analytical expression for the average number of received packets in delay-constrained networks, whereas [8] measures the complexity of different fountain code relaying strategies. In [13], the authors consider a multihop scenario with a single node at each relay stage, and analyze the probability distribution of the transmission time in the presence of interference from randomly distributed nodes. However, in [5], [8], [12], [13], multihop communications take place through a single, fixed path determined a priori, rather than considering a flooding scenario as we do in this paper.

In [14], erasure codes have been applied to multicast transmissions of short duration in a single-hop cellular scenario. Fountain codes have also been shown to improve data dissemination [15], MAC layer communications [16] and reliable transport [4] in underwater networks. Random linear packet coding is applied to single-hop broadcast networks in [17]. Transmit power adjustment policies are proposed based either on the performance of the worst link or on the average performance of all links. In [18], in order to increase the reliability of packet delivery over multihop unicast connections, a node determines the lost packets thanks to an implicit acknowledgment procedure and retransmits them at a later time if needed. In [19], restricted flooding is proposed, where a node transmits with probability p after receiving the m th copy of a message, where p is selected based on percolation theory.

Several approaches have been proposed to mitigate the overhead coming from the excessive transmission redundancy. In [20], a network coding-based approach is discussed to find the minimum forwarding probability required to achieve a given outreach probability over random graphs. The authors in [21], [22] discuss dynamic probabilistic broadcasting schemes in MANETs: the retransmission probability is adjusted based on the number of neighboring nodes (the higher the number of neighboring nodes, the lower the rebroadcasting probability). In [23], adaptive probabilistic flooding is applied to path discovery in multipath routing: the retransmission probability of advertisement messages is progressively reduced to limit the flooding process. A similar scheme is also considered

in [24] to reduce transmissions in a vehicular ad hoc network, where the rebroadcasting probability is tied to the vehicle speed. Recently, [25] proposed that a node tunes network coding based on packet reception probability estimates for neighboring nodes, in order to maximize the aggregate number of source packets decoded by all neighbors while limiting the dissemination delay. These estimates are evaluated based on the link quality correlation among the neighboring nodes.

In [26], lower bounds for the required number of packet retransmissions at the MAC layer are derived in order to support broadcast with and without applying network coding techniques. In [27], a reliable broadcast transmission approach based on random linear network coding is considered, in which all source packets are initially transmitted, and once the ACK/NACK storm has finished, the source transmits network-coded packets to improve the performance of the worst receiver. In [28], the authors proposed a low-complexity and energy-efficient fractional transmission scheme (FTS) broadcasting approach that employs fountain codes over a multihop wireless network. Based on hop distance, a fraction of the encoded packets to be transmitted to a particular node is assigned among its neighboring nodes. If the sum of the received fractions exceeds one, then the assigned fractions can be adapted to reduce the number of redundant transmissions. After successful decoding, a node starts transmitting additional encoded packets.

Unlike previous work, in this paper we study the interaction of flooding and fountain codes via a fundamental approach. We start with a mathematically tractable model of the network performance that is proven to match realistic simulations quite accurately. We employ the analysis to identify desirable working points in terms of retransmission overhead and probability of success, and finally proceed to propose heuristic policies that can approximate the desired probabilistic forwarding behavior, in terms of both probability of success and overhead. Unlike in such works as [18], [19], [23], our practical policies rely on decisions that are made by each node in a fully distributed manner. Moreover, unlike [20], [25]–[27], our policies do not rely on further network coding, hence the source is in control of the behavior of the flooding process.

III. SCENARIO DESCRIPTION, ANALYTICAL MODEL, AND PRACTICAL FLOODING POLICIES

We consider a multihop network scenario, where a source S has m packets to send to a given destination node D . To protect the communication from packet losses, the source encodes these m packets into $M > m$ packets using a random fountain code over the Galois field of size η , \mathbb{F}_η , $\eta \geq 2$. We define $E = M - m$ as the number of redundancy packets injected into the network to favor the decoding process at the destination, and to combat packet loss over the network links. All packets are forwarded towards the destination through a multihop network where we assume that there are r relay stages between S and D , and each stage i consists of N_i relays. This way, the shortest path joining S and D requires $r + 1$ hops. In addition, we assume a Bernoulli error model, whereby the probability of packet error over each link is

constant and equal to p . While this assumption is required to make the problem tractable, we will show in Section IV that it is a very good approximation for more realistic link error models.¹ In the following three subsections, we will model the forwarding process from S to D , by considering different types of flooding. We will derive the probability that D can decode the fountain code, as well as the number of transmissions that the network carries out to advance the fountain code packets towards the destination. Specifically, Section III-A focuses on a baseline, unrestricted flooding paradigm; Section III-B describes distributed and tunable forwarding policies that achieve a good tradeoff between the probability of decoding success at D and the number of transmissions in the network; Section III-C extends the analytical model to a form of restricted flooding, which is leveraged to provide insight on the practical policies; finally, Section III-D comments on the fundamental interplay between the Galois field size employed for the fountain code and the network performance.

A. Unrestricted flooding model

We start by assuming that the forwarding paradigm adopted in the network is a baseline full flooding, i.e., a multihop forwarding process where every unique packet is transmitted once by the source, and forwarded exactly once by each node that correctly receives it. Possible collisions among the transmissions by two or more relays of either the same or different stages can be avoided in several ways. Feasible measures include channel sensing, fine-tuned backoff times and transmission pipelining [3], and typically introduce additional delays. However, we note that applications relying on flooding to convey data through a network are most likely concerned about the probability of successful delivery than about the time it takes to deliver a packet. Therefore, in this paper, we focus mainly on the probability of success, discuss a model to measure it, and present effective distributed ways to achieve successful packet delivery while limiting the number of transmissions in the network. In this light, a collision avoidance mechanism that introduces some delay is not a restraining assumption.² For reference, a scheme of our network scenario is provided in Fig. 1.

We recall that, assuming at least $\ell \geq m$ packets are received by D , the probability $P_{e,D}$ that D cannot decode the fountain code is upper-bounded by [29]

$$P_{e,D} \leq \eta^{-(\ell-m)}, \quad \ell \geq m \quad (1)$$

¹It could be argued that allowing nodes at generic stage i to receive transmissions from farther stages rather than just $i-1$ would help improve the probability of correct reception at stage i and, as a consequence, at the destination as well. In our extensive set of simulations (not shown here due to lack of space) we confirmed that this is not the case, except in a very restricted set of scenarios with benign channels (i.e., very low p) and very few nodes in each stage. The latter is typical of standard multi-hop routing scenarios, which are not the focus of this paper. In all other cases, the gain is negligible because either a) the probability of error from stage $i-2$ to stage i is very high or b) the flooding process compensates for transmission errors and is thus the main reason behind successful reception at stage i .

²A possible alternative for collision modeling would be to incorporate the probability of collision into the link error probability p . This entails the analysis of the interplay between MAC- and routing-level performance, and is left as a future extension.

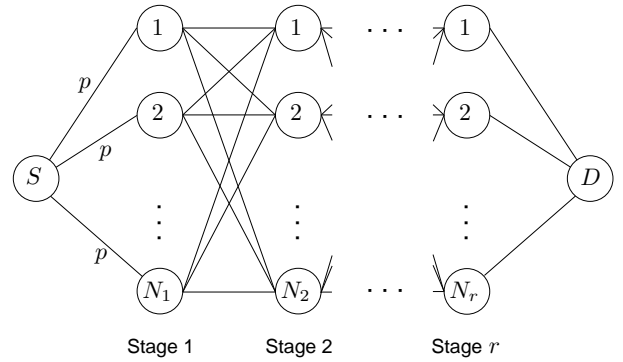


Figure 1. Reference scenario for a multihop network with multiple relay stages between a source S and a destination D . The probability of error over any link is equal to p .

where η is the size of the Galois field over which the code is designed, e.g., $\eta = 2$ for a binary fountain code. If $\ell < m$, we set $P_{e,D} = 1$.

Based on the assumptions above, we now derive the probability of decoding error at D . Assume for the moment that $r = 1$, i.e., there is only one relay stage between S and D , and the length of the path from S to D is 2 hops. The event that a given encoded packet does not reach D is equivalent to the event that no relay receives the packet correctly, or that all relays that received the packet fail to forward it further to D . The probability of this event is

$$q_1 = \sum_{j_1=0}^{N_1} \binom{N_1}{j_1} p^{j_1} (1-p)^{N_1-j_1} p^{N_1-j_1} = (p + (1-p)p)^{N_1}, \quad (2)$$

where we recall that p is the link error probability. The argument of the summation means that if $N_1 - j_1$ nodes out of N_1 receive the packet correctly, all these nodes fail in forwarding it to D over the second hop. We note that the number of nodes that fail to receive a packet at relay stage 1 follows a binomial distribution $\mathcal{B}(N_1, p, j_1)$ with parameters N_1 and p , hence its average is $N_1 p$. Considering the degenerate cases where $p = 1, j_1 = N_1$ (all nodes fail to receive if the link error probability is identically 1) and when $p = 0, j_1 = 0$ (all nodes are successful if the link error probability is identically 0), we note that $\mathcal{B}(N_1, p, j_1)$ should be extended to yield 1 in both cases. We therefore define $\bar{\mathcal{B}}(N_1, p, j_1) = 1$ if $p = 1, j_1 = N_1$ or $p = 0, j_1 = 0$, and $\bar{\mathcal{B}}(N_1, p, j_1) = \binom{N_1}{j_1} p^{j_1} (1-p)^{N_1-j_1}$ otherwise.

Assume now that $r = 2$, i.e., there are two relay stages between S and D , respectively, containing N_1 and N_2 relays. The probability that a packet is correctly received by any given number of relays $1, 2, \dots, N_2$ of stage 2 depends on the number of nodes that successfully received the same packet at stage 1. Hence the probability that the packet does not reach D is

$$q_2 = \sum_{j_1=0}^{N_1} \sum_{j_2=0}^{N_2} \bar{\mathcal{B}}(N_1, p, j_1) \bar{\mathcal{B}}(N_2, p^{N_1-j_1}, j_2) p^{N_2-j_2}, \quad (3)$$

where j_1, j_2 track the number of relays that failed the reception of the packet at stages 1 and 2, respectively. We note that, for

a given value of j_1 , the probability of failing to advance a packet from stage 1 to any node in stage 2 is $p^{N_1-j_1}$. In the general case, there are r relay stages between S and D , and the expression for the probability of error can be given as

$$q_r = \sum_{j_1=0}^{N_1} \sum_{j_2=0}^{N_2} \cdots \sum_{j_r=0}^{N_r} \bar{\mathcal{B}}(N_1, p, j_1) \bar{\mathcal{B}}(N_2, p^{N_1-j_1}, j_2) \times \cdots \times \bar{\mathcal{B}}(N_r, p^{N_{r-1}-j_{r-1}}, j_r) p^{N_r-j_r}, \quad (4)$$

where j_1, j_2, \dots, j_r are the numbers of relays that failed the reception of the packet at the first, second, up to the r th stage, respectively, and therefore $(N_{r-1} - j_{r-1})$ is the number of nodes which received the packet correctly at the $(r-1)$ th relay stage. Finally, by virtue of the approximate formula in (1), the probability that D fails to decode the fountain code is upper-bounded by 1 whenever D correctly receives only m or fewer coded packets. Conversely, the probability of decoding failure is upper-bounded as in (1) whenever at least $m+1$ packets are received. Therefore, we have the following upper bound:

$$P_{\text{fail}}(q_r) \leq \sum_{k=0}^m \bar{\mathcal{B}}(M, 1 - q_r, k) + \sum_{k=m+1}^M \bar{\mathcal{B}}(M, 1 - q_r, k) \eta^{-(k-m)} \quad (5)$$

which in non-degenerate cases can be explicitly re-written as

$$P_{\text{fail}}(q_r) \leq \sum_{k=0}^m \binom{M}{k} q_r^{M-k} (1 - q_r)^k + \sum_{k=m+1}^M \binom{M}{k} q_r^{M-k} (1 - q_r)^k \eta^{-(k-m)}, \quad (6)$$

where $k - m$ is the number of extra packets received by D .

The formula in (4) does not admit a closed-form expression, and neither do (5) and (6). However, we will now illustrate a conveniently compact way to write (4), which will be leveraged to compute other metrics. We start by grouping the probabilities that a given number of relays fail packet reception at a given stage into matrices. To do so, we define the following probabilities

$$p_{S,j} = \bar{\mathcal{B}}(N_1, p, j), \quad p_{i,jk} = \bar{\mathcal{B}}(N_i, p^{N_{i-1}-j}, k), \quad p_{D,k} = p^{N_r-k}. \quad (7)$$

In particular, $p_{S,j}$ is the probability that j nodes at the first relay stage fail to receive the packet from S ; $p_{i,jk}$ is the probability that k nodes at the i th relay stage fail to receive the packet given that j nodes failed to receive it at stage $i-1$; $p_{D,k}$ is the probability that the destination D fails to receive the packet, given that k nodes (out of the available N_r nodes at stage r) failed to receive it at stage r . We now define the column vectors $\mathbf{p}_S = [p_{S,0} \ p_{S,1} \ \cdots \ p_{S,N_1} \ \mathbf{0}_{1 \times (N_{\max} - N_1)}]^T$, $\mathbf{p}_D = [p_{D,0} \ p_{D,1} \ \cdots \ p_{D,N_r} \ \mathbf{0}_{1 \times (N_{\max} - N_r)}]^T$, and the matrix

$$\mathbf{P}_i = \left[\begin{array}{ccc|c} p_{i,00} & \cdots & p_{i,0N_i} & \mathbf{0}_{(N_{i-1}+1) \times (N_{\max} - N_i)} \\ \vdots & \ddots & \vdots & \\ p_{i,N_{i-1}0} & \cdots & p_{i,N_{i-1}N_i} & \mathbf{0}_{(N_{\max} - N_{i-1}) \times (N_{\max} + 1)} \end{array} \right], \quad (8)$$

where $N_{\max} = \max_i N_i$, and we remark that $p_{i,N_{i-1}N_i} = 1$ is the only non-zero element of the N_i th row. Eq. (4) can now be rewritten as

$$q_r = \mathbf{p}_S^T \left(\prod_{i=1}^r \mathbf{P}_i \right) \mathbf{p}_D, \quad (9)$$

where it is intended that $\mathbf{P}_1 = \mathbb{I}_{N_{\max}+1}$, the $(N_{\max} + 1) \times (N_{\max} + 1)$ identity matrix. The value of q_r thus derived can be finally plugged into (6) to obtain an upper bound to the probability of decoding the fountain code at D .

We illustrate the formulation above via a simple example. Consider a scenario with $r = 2$ and $N_1 = N_2 = 1$. In this case, $\mathbf{p}_S = [1 - p \ p]^T$, $\mathbf{p}_D = [p \ 1]^T$, and matrix \mathbf{P}_1 is given as

$$\mathbf{P}_2 = \begin{bmatrix} \bar{\mathcal{B}}(1, p, 0) & \bar{\mathcal{B}}(1, p, 1) \\ \bar{\mathcal{B}}(1, 1, 0) & \bar{\mathcal{B}}(1, 1, 1) \end{bmatrix} = \begin{bmatrix} (1-p) & p \\ 0 & 1 \end{bmatrix}. \quad (10)$$

Finally we have

$$q_2 = \mathbf{p}_S^T \mathbf{P}_1 \mathbf{P}_2 \mathbf{p}_D = [1 - p \ p] \begin{bmatrix} 1-p & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = (1-p)^2 p + (1-p)p + p. \quad (11)$$

For convenience note that, if $r = 2$, (3) can be rewritten as

$$q_2 = \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2} \binom{N_1}{j_1} p^{j_1} (1-p)^{N_1-j_1} \times \binom{N_2}{j_2} (p^{N_2-j_1})^k (1-p^{N_1-j_1})^{N_2-j_2} p^{N_2-j_2} + p^{N_1}, \quad (12)$$

where the last term conveys the fact that if all nodes at stage 1 fail to receive a packet from the source, the destination will also fail to receive it. For $N_1 = N_2 = 1$ and $r = 2$, (12) expands to

$$q_2 = (1-p)^2 p + (1-p)p + 0 + p, \quad (13)$$

where the four terms are computed for $(j_1, j_2) = (0, 0), (0, 1), (1, 0)$ and $(1, 1)$, respectively. As expected, (11) equals (13), and both equal $1 - (1-p)^3$, i.e., the probability of error over a 3-hop path, where the error probability over each link is p .

Along the same line of (9), we now compute the average number of transmissions that all nodes carry out to advance a given packet towards the destination (regardless of whether the packet actually reaches the destination or not). Focus on relay stage i and assume that j_{i-1} nodes failed the reception of the packet at stage $i-1$. The conditional number of transmissions that take place at stage i is equal to the number of nodes that received the transmission correctly. Its average over the distribution of the number of successful nodes at stage i can be found as

$$u_{i|j_{i-1} \dots j_1} = \sum_{j_i=0}^{N_i} \bar{\mathcal{B}}(N_i, p^{N_{i-1}-j_{i-1}}, j_i) (N_i - j_i) = N_i (1 - p^{N_{i-1}-j_{i-1}}), \quad (14)$$

Eq. (14) must be averaged over $j_{i-1}, j_{i-2}, \dots, j_1$ to yield the unconditional average number of transmissions at the i th relay

stage. Call this quantity u_i . To compute the average of (14), we can replicate the structure of (9) by first defining the column vectors

$$\mathbf{t}_i = [N_i \quad N_i - 1 \quad \dots \quad 1 \quad 0]^T. \quad (15)$$

Thus, $u_i = \mathbf{p}_S^T (\prod_{j=0}^{i-1} \mathbf{P}_j) \mathbf{t}_i$, and the average total number of transmissions is found as

$$T_{\text{avg}} = 1 + \sum_{i=1}^r u_i. \quad (16)$$

We remark that $u_1 = N_1(1-p)$ and $u_2 = N_2(1-(p+(1-p)p)^{N_1})$, after which no closed-form expression can be obtained for the remaining u_i s.

If we assume that each stage hosts the same number of relays N , then $\mathbf{P}_i = \mathbf{P} \forall i$, where \mathbf{P} is an $(N+1) \times (N+1)$ matrix, and we can simplify (9) to yield

$$q_r = \mathbf{p}_S^T \mathbf{P}^{r-1} \mathbf{p}_D, \quad (17)$$

and $T_{\text{avg}} = 1 + \sum_{i=1}^r \mathbf{p}_S^T \mathbf{P}^{i-1} \mathbf{t}_i$.

B. Distributed Flooding Policies

In many cases, typically if the link error probability is sufficiently low, the multiplicity of the flooding process is excessive, and leads to several packet replicas being uselessly transmitted, without noticeably increasing the success probability at the destination. It is therefore appropriate to design practical flooding policies that attempt to reduce this number of transmissions. Specifically, each node should decide locally whether to retransmit a given packet or not, without explicit coordination mechanisms.

We achieve this by allowing the nodes to overhear nearby traffic to understand how many relays have retransmitted a given packet at each stage. We design two policies for this purpose: the first is named Predetermined Restriction (PR), and the second is Adaptive Restriction (AR). PR is aimed at enforcing a maximum number of transmitters per stage. Conversely, the objective of AR is to keep the flooding process running by adapting to the network conditions: if many successful retransmissions are detected, fewer nodes will tend to transmit; if overheard traffic suggests that a given packet is not being correctly advanced to subsequent stages, more nodes will act to support the flooding process.

For all policies, we assume the presence of a MAC protocol that avoids collisions between retransmissions by different relays. In practice, this can be achieved with very high probability by having relays back off before performing a retransmission, where the length of the backoff interval is drawn within a sufficiently long backoff window, or by loosely synchronizing subsequent relay stages in order to make transmission pipelining possible [3]. To fix ideas, in the following we assume the case of a backoff-based MAC protocol. We also assume that each node listens to the channel during the backoff period and, when its own backoff timer expires, the decision to transmit is made based on the policies described in the following subsections.

1) *Flooding with Predetermined Restriction (PR)*: The PR flooding policy prescribes that a node at a given stage should avoid retransmitting a packet if it overhears at least one retransmission of the same packet by any other node at the same stage, and the SNR of this transmission is greater than a prescribed value θ_r . Therefore, we can approach some desired average maximum number of relays per stage, \hat{N}^{res} , by varying θ_r .

To formalize the above ideas, assume, without loss of generality, that the nodes are ordered and numbered increasingly in subsequent relay stages, i.e., those of the first stage are numbered from 1 to S_1 , those of the second stage from $S_1 + 1$ to S_2 , and those of the i th stage from $S_{i-1} + 1$ to S_i , where $S_k = \sum_{\ell=1}^k N_\ell$. As a packet is forwarded from stage $i-1$ to stage i , the nodes at stage i start entering backoff periods to relay the packet, and keep listening for retransmissions in the meantime. As a node, say j , is ready to retransmit a packet from stage i , it will check if there exists at least one additional relay $S_{i-1} + 1 \leq k \leq S_i$, for which $\gamma_{kj} \geq \theta_r$, where γ_{kj} is the SNR of node k 's transmission as measured by node j . In this case the node will refrain from transmitting, otherwise it will relay the packet to the next stage. We note that, by increasing θ_r , fewer overheard retransmissions will meet the SNR threshold; in turn, the probability that a node is silenced decreases, and the average number of relays per stage increases. Conversely, by decreasing θ_r , a node can be silenced on average by a larger number of retransmitted packets, hence the probability that the node will refrain from retransmitting increases. As a result, this will limit the average number of relays per stage.

2) *Flooding with Adaptive Restriction (AR)*: The AR policy is designed to achieve a high probability of success while at the same time reducing the overhead of the full flooding mechanism. Focusing on a given packet with sequence number n , the AR policy proceeds over the following two steps.

Step 1: Perform a relaying step according to the PR policy. At a generic stage i , this will result in some nodes having relayed the packets, whereas other nodes will have refrained from doing so according to the rules of the PR policy (see Section III-B1). For packet n , call $C_i^{(s)}(n)$ the set of the nodes silenced at stage i : we note that the set can vary for different packets, hence for different values of n . All silenced nodes in $C_i^{(s)}(n)$ remain in a listening state, attempting to overhear packet n being forwarded further downstream by nodes at stage $i+1$.

Step 2: For stage i , any node that hears packet n being transmitted at least once from stage $i+1$ to stage $i+2$ exits the flooding process for packet n permanently. Conversely, the nodes that could not hear packet n being forwarded from stage $i+1$ to stage $i+2$ assume that the forwarding process might have been unsuccessful. Therefore, they break silence and relay packet n from stage i to stage $i+1$.

We note that this further transmission step may result in additional nodes that receive the packet successfully at stage $i+1$. In the meantime these nodes will have overheard other transmissions in their neighborhoods, and will also decide to transmit or not according to steps 1 and 2 of the AR policy. However, in no event will a node forward a received packet

more than once. Therefore, the number of transmissions carried out under the AR policy will be always less than or equal to that of unrestricted flooding. Both the PR and AR policies work based on the idea that the forwarding operations could take advantage of some inherent MAC-level mechanisms that are typically present in MAC protocols for wireless networks, including idle listening and backoff. Instead of restricting such operations to the role of informing the MAC logic, we leverage on them to optimize the flooding performance by achieving a good tradeoff between the probability of decoding success at D and the number of transmissions. Further insight on this aspect is provided in our performance evaluation in Section IV.

C. Restricted flooding model

Both the PR and the AR policies act by reducing the total number of relays per stage, either in a fixed (PR) or in an adaptive fashion (AR). An analytical framework to assess the performance of these policies can be provided by extending the model of Section III-A to include restricted flooding. Restricted flooding is generally defined as a flooding policy where some nodes refrain from forwarding a packet even though they received it correctly. In this paper, we model restricted flooding as a limit N^{res} on the maximum number of nodes that relay a packet at each stage.³ To simplify the analysis, we assume that the number of successful nodes is known to all relays of a given stage, and that some form of arbitration takes place, in order to make sure that up to N^{res} relays transmit. Note that this assumption was already relaxed in the design of our proposed heuristic policies (Section III-B).

Under the assumptions above, the analysis carried out so far offers a straightforward way to model restricted flooding. Assume that we have the same number of relays at each stage, which is the case, e.g., in (17): the analysis can be easily extended to the more general case of Eq. (9). Restricting the maximum number of relays to N^{res} means that whenever more than N^{res} relays correctly receive a packet at a given stage, only N^{res} of them will actually retransmit. This can be modeled by setting the first $N - N^{\text{res}} + 1$ lines of matrix \mathbf{P} in (17) to be equal to the vector

$$\mathbf{p}^{\text{res}} = [p_{(N-N^{\text{res}})0} \quad p_{(N-N^{\text{res}})1} \quad \cdots \quad p_{(N-N^{\text{res}})N}]^T \quad (18)$$

to yield

$$\mathbf{P}^{\text{res}} = \begin{bmatrix} & & (\mathbf{p}^{\text{res}})^T & & \\ & & \vdots & & \\ & & (\mathbf{p}^{\text{res}})^T & & \\ p_{(N-N^{\text{res}}+1)0} & \cdots & p_{(N-N^{\text{res}}+1)N} & & \\ \vdots & \ddots & \vdots & & \\ p_{N0} & \cdots & p_{NN} & & \end{bmatrix}. \quad (19)$$

In the same vein, \mathbf{p}_D becomes $\mathbf{p}_D^{\text{res}} = [p_{D,N-N^{\text{res}}} \quad \cdots \quad p_{D,N-N^{\text{res}}} \quad p_{D,N-N^{\text{res}}+1} \quad \cdots \quad p_{D,N}]^T$, so that $q_r^{\text{res}} = \mathbf{p}_S^T (\mathbf{P}^{\text{res}})^{r-1} \mathbf{p}_D^{\text{res}}$, and the probability that D

³Note that, in this model, N^{res} denotes the (exact) maximum number of nodes that will forward a correctly received packet at any stage, as opposed to the *average* number of nodes allowed to forward a packet at each stage in the PR and AR policies, which was denoted as \bar{N}^{res} in Section III-B.

fails to decode the fountain code is $P_{\text{fail}}(q_r^{\text{res}})$. To compute the average number of transmissions in the network, we first define the column vector

$$\mathbf{t}_D^{\text{res}} = [N - N^{\text{res}} \quad \cdots \quad N - N^{\text{res}} \quad N - N^{\text{res}} - 1 \quad \cdots \quad 0]^T, \quad (20)$$

and finally the average number of transmissions is found as

$$T_{\text{avg}}^{\text{res}} = 1 + \sum_{i=1}^r \mathbf{p}_S^T (\mathbf{P}^{\text{res}})^{i-1} \mathbf{t}_D^{\text{res}}. \quad (21)$$

D. Selecting the Galois field size η

Before proceeding to present analysis and simulation results in Section IV, it is useful to choose the size of the Galois field over which the flooded fountain code is designed. We do this by evaluating the interplay between the Galois field size and $1 - P_{\text{fail}}$. In fact, a typical fountain code packet contains an encoding vector that defines how many packets have been linearly combined, and with which coefficients. Designing the fountain code over a small Galois field (at the minimum, \mathbb{F}_2) keeps the encoding vector short at one bit per encoded packet; however, in this case Eq. (1) dictates that the probability of successful decoding at the destination becomes lower for a fixed number of redundancy packets E . Conversely, a large Galois field (e.g., \mathbb{F}_{256} as typically assumed in rateless code design) provides a higher probability of success for equal E , but at the same time it increases the size of each coded packet. Given that we consider the flooding of fountain codes through a network, where each packet is expected to be retransmitted several times by different nodes at each hop, we need to find a good tradeoff between the overhead yielded by the encoding vector size, and that yielded by the number of extra packets.

Assume that each packet has the following structure: a fixed overhead that represents generic protocol information (e.g., Time-To-Live and version fields, flags, etc.) of size α_{fix} ; a sequence number that distinguishes different sequences of m source packets, of size α_{seq} ; and an encoding vector of $m \times \xi$ bits, where $\xi = \log_2 \eta$, and η is defined in (1); finally, a payload of L bits. We calculate the transmission overhead affecting a sequence of m source packets as

$$O_{\text{tx}} = \frac{(m + E)(\alpha_{\text{fix}} + \alpha_{\text{seq}} + m\xi) + E \cdot L}{(m + E)(\alpha_{\text{fix}} + \alpha_{\text{seq}} + m\xi + L)}. \quad (22)$$

Fig. 2 shows a plot of $1 - P_{\text{fail}}(q_r)$ vs. O_{tx} for different values of ξ , E and $1 - p$. The packet parameters are $\alpha_{\text{fix}} = 24$ bits, $\alpha_{\text{seq}} = 8$ bits, $L = 256$ bits. For each value of $1 - p$, a set of markers of the same color is shown; the set is spanned from bottom-left to top-right by increasing ξ from 1 to 8. Three sets of curves are shown, respectively for $E = 1$, $E = 4$ and $E = 8$: a curly brace encompasses the horizontal span of each set. As expected, the probability of success increases with increasing ξ , and with increasing E for fixed ξ . However, both improvements come at the cost of an increase in O_{tx} . While the increase of O_{tx} with E is expected and depends on the configuration of the code, the value of ξ can still be optimized. In particular, we observe that $1 - P_{\text{fail}}(q_r)$ remains almost constant after $\xi = 4$. Therefore, we will fix this value throughout the rest of this paper.

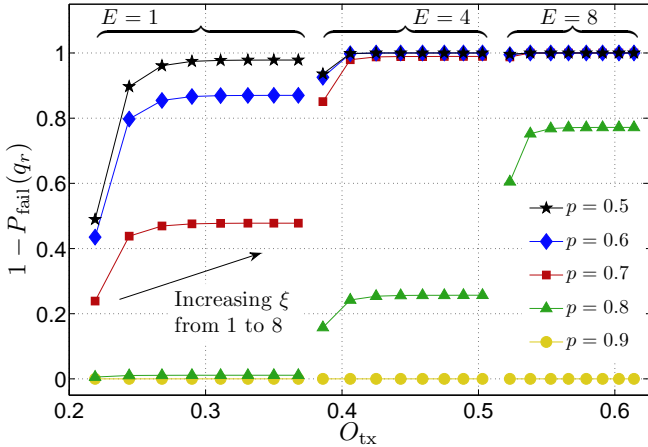


Figure 2. $1 - P_{\text{fail}}(q_r)$ vs. O_{tx} for different values of ξ , E and of the probability of link error p . For each value of p , a set of markers of the same color is shown; the set is spanned from bottom-left to top-right by increasing ξ from 1 to 8. The results show no significant improvement in $1 - P_{\text{fail}}(q_r)$ for $\xi \geq 4$.

IV. NUMERICAL RESULTS

A. Simulation scenario

We now present some results related to the flooding of fountain codes through a multihop network. We will show that our analytical model, albeit based on a simple uniform Bernoulli error process, matches simulations based on a more realistic Rayleigh fading propagation model with good accuracy, provided that an appropriate map between the error probability p of the Bernoulli model and the average probability of error in the Rayleigh fading scenario is used.

The simulations are set up as follows. We represent each relay stage as a set of nodes whose position is drawn at random within a cell of size $20 \text{ m} \times 60 \text{ m}$, where the centers of any two nearest cells are $\Delta = 60 \text{ m}$ apart. This scenario makes it possible to test that our analysis is still valid even after removing one of the assumptions of Section III, namely that any two nodes belonging to subsequent relay stages are always connected. The source S and the destination D are located, respectively, at the center of the first and last cells. We assume that all nodes transmit with a power P_T . Therefore, if two nodes i and j are located at a distance d_{ij} , the Signal-to-Noise Ratio (SNR) γ_{ij} of the link between the two nodes can be computed as

$$\gamma_{ij} = \rho_{ij} P_T d_{ij}^{-\alpha} / P_N \quad (23)$$

where α is the path-loss exponent, ρ_{ij} is an exponentially distributed power fading coefficient of average value 1 (which corresponds to a Rayleigh fading amplitude) representing the fading realization on the link from i to j , and P_N is the noise power. We assume that a transmission from i to j is successful if and only if $\gamma_{ij} \geq \theta$, where θ is a minimum SNR threshold.

In order to compare the Rayleigh fading simulations with the Bernoulli link error model presented in Section III, we map the parameter p of the Bernoulli model to the average error probability computed via a stochastic geometry argument. Consider two nodes i and j , where i belongs to stage ℓ ,

$\ell = 1, 2, \dots, r - 1$ and j to stage $\ell + 1$. Without loss of generality, let $\ell = 1$, i.e., focus on the first and second relay stages. Call (x_i, y_i) and (x_j, y_j) the coordinates of nodes i and j , and assume that they take values in $[0, X_i]$, $[0, Y_i]$, $[\Delta, \Delta + X_j]$, $[0, Y_j]$, respectively. In our simulation scenario, $X_i = X_j = 20 \text{ m}$, $Y_i = Y_j = 60 \text{ m}$ and $\Delta = 60 \text{ m}$. The map is obtained as follows:

$$p = \int_0^{X_i} dx_i \int_0^{Y_i} dy_i \int_{\Delta}^{\Delta + X_j} dx_j \int_0^{Y_j} dy_j \int_0^{+\infty} d\rho_{ij} \left(\frac{e^{-\rho_{ij}}}{A_i A_j} \mathbb{1} \left[\frac{\rho_{ij} P_T d_{ij}^{-k}}{P_N} < \theta \right] \right), \quad (24)$$

where $A_i = X_i Y_i$, $A_j = X_j Y_j$, $e^{-\rho} / (A_i A_j)$ is the joint distribution of fading and of the locations of nodes i and j , and $\mathbb{1}[\cdot]$ denotes the indicator function, which returns 1 whenever the argument is true. We remark that (24) computes the average value of the indicator function over all random parameters that concur to the computation of the link error probability. Such an average is by definition the probability of a Bernoulli event [30, Section 3.1], which is fully in line with our Bernoulli link error probability model. In the following, we set $\alpha = 2$, $P_T = 34 \text{ dBm}$ and $P_N = -17 \text{ dBm}$, we consider the network to be composed of $r = 5$ relay stages, and we assume that there are exactly 10 nodes in any stage, i.e., $N_1 = \dots = N_r = 10$. Different probabilities of success are obtained by varying θ between 0 and 20 dB. All results are averaged over 2000 random draws of the nodes' positions, and over 10 different fading realizations for each position.

In the next Subsection, we will discuss the performance of flooded fountain codes in the presence of both full and restricted flooding. In Subsection IV-C, we will discuss the performance of our proposed practical policies, while in subsection IV-D, we will compare these policies with the restricted flooding model. In Section V, we show the flexibility of our formulation by extending the analysis to the case of a ‘‘helper’’ node injecting additional redundancy in the network, and discuss the optimal placement of that node.

B. Performance of fountain codes under unrestricted flooding

We start from Fig. 3, which shows $1 - P_{\text{fail}}(q_r)$ vs. p in the case of unrestricted flooding, by comparing our analytical model against Rayleigh fading simulation outcomes, for two different values of E . We observe that the probability of decoding success at the destination for $E = 8$ remains practically equal to 1 for $p \leq 0.75$, after which it sharply falls and becomes practically 0 for $p \geq 0.85$ due to the excessive number of forwarding errors. The anticipated effect of a lower number of redundancy packets, for $E = 2$, is that the probability of success is practically 1 only for $p \leq 0.5$, and the transition to 0 is also smoother. In both cases, Fig. 3 shows a very good agreement between the simulations and the analytical model.

A second metric of interest for characterizing the network behavior is the average number of nodes that successfully receive a given packet at each relay stage for different values of p . This metric is depicted in Fig. 4. The results show that the number of successful nodes at each stage increases for

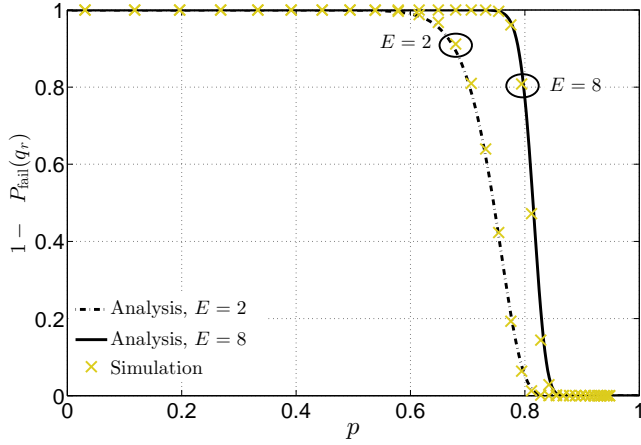


Figure 3. Success probability $1 - P_{\text{fail}}(q_r)$ vs. p for $m = 10$, $r = 5$ and $N = 10$, for varying number of redundancy packets, E . The probability that the destination can decode the fountain code, mainly driven by E , dominates the performance for sufficiently low values of p .

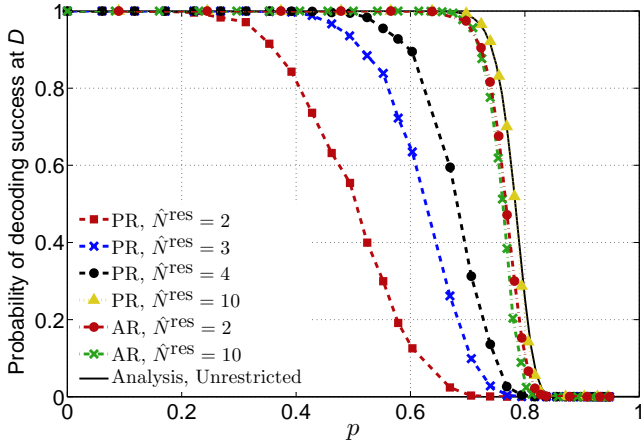


Figure 5. Probability of decoding success at D as a function of p for different combinations of \hat{N}^{res} and N^{res} , for the PR and the AR policies, compared to the analysis for unrestricted flooding.

decreasing p and that, even for moderately high values of p (up to 0.6 in Fig. 4), practically all $N_r = 10$ nodes in the last relay stage have correctly received the packet. Conversely, $p = 0.8$ reduces the number of correct transmissions, and it is difficult for the flooding process to involve several nodes before the packet traverses the last relay stage. In all cases, the simulations match the analysis very well.

One of the main conclusions from the previous results is that $1 - P_{\text{fail}}(q_r)$ remains equal to 1 even for significantly high values of p . This further motivates the design of practical policies that achieve the probability of success of unrestricted flooding, while requiring fewer transmissions. The following subsection presents the performance evaluation of the PR and AR policies, introduced in Section III-B1 and III-B2.

C. Performance of the PR and AR policies

The behavior of the PR and AR policies is simulated by assuming the Rayleigh fading propagation model discussed

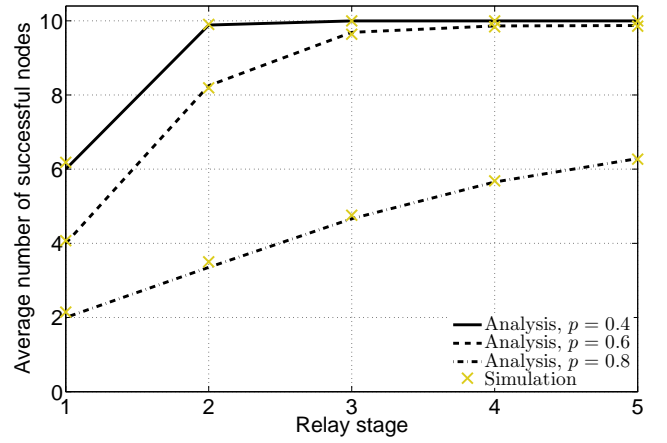


Figure 4. Average number of nodes that received a given packet correctly as a function of the relay stage (from 1 to 5) for different values of p . For sufficiently high link error probability ($p = 0.8$), in every stage there exist some nodes that fail to receive a given packet.

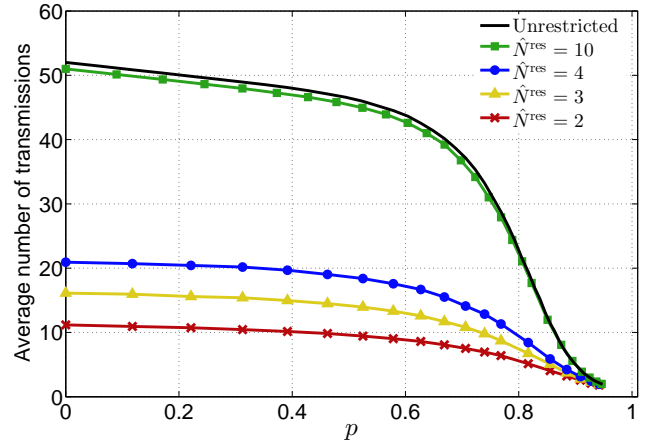


Figure 6. Average number of transmissions as a function of p for the PR policy for different values of \hat{N}^{res} , compared to the analysis for unrestricted flooding.

in Section IV-A, and compared to the analytical model for unrestricted flooding via the map for p given in (24).

Fig. 5 shows a plot of the probability of correct decoding at D against p obtained via the simulation of the PR and AR policies, compared to the unrestricted flooding analysis. For the latter, this is equivalent to $1 - P_{\text{fail}}(q_r)$. The PR and AR curves are plotted for different values of \hat{N}^{res} . There are $N = 10$ nodes in all relay stages, and the number of redundancy packets transmitted by the source S is fixed to $E = 4$. Considering the PR policy, we observe that increasing the \hat{N}^{res} parameter (by choosing θ_r as explained in Section III-B1) correspondingly increases the number of relays that are allowed to re-forward a packet at each stage. For example, according to our simulation setup, setting $\theta_r \approx 40$ dB corresponds to $N^{\text{res}} = 8$, whereas $\theta_r \approx 21$ dB corresponds to $N^{\text{res}} = 2$. Increasing \hat{N}^{res} by increasing θ_r progressively improves the probability of success as a function of p , until it finally matches that of unrestricted flooding for $\hat{N}^{\text{res}} = 10$. Contrary to the PR policy, the AR policy is adaptive. Posing

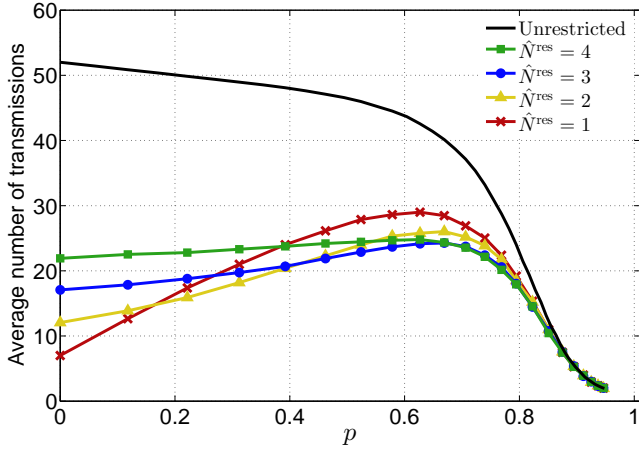


Figure 7. Average number of transmissions for a single packet carried out by the AR policy as a function of p for different values of \hat{N}^{res} , compared to the number of transmissions of unrestricted flooding.

restrictions on the number of forwarders via \hat{N}^{res} makes the policy react by allowing more silenced relays to retransmit, in case they fail to hear relays at the next stage re-forward a packet. Therefore, as we observe from Fig. 5, the success probability of the AR policy is always comparable to that of unrestricted flooding. This proves the effectiveness of the policy at compensating for forwarding errors.

It is interesting to evaluate the behavior of the PR and AR policies in terms of the number of transmissions performed per packet carried through the network. This metric is depicted in Fig. 6 for PR, and shows the expected behavior that a higher value of \hat{N}^{res} results in more relays being activated per stage, hence in more transmissions. For the same configurations shown in Fig. 6, we observe a steady increase in the number of transmissions. For $\hat{N}^{\text{res}} = 10$, PR transmits practically as many packets as unrestricted flooding. A different behavior is observed for the average number of transmissions per packet of the AR policy as a function of p (Fig. 7). When $p < 0.2$, the low probability of error makes the choice of $\hat{N}^{\text{res}} = 4$ very inefficient, as in this condition it is highly likely that the nodes will overhear other transmissions, refrain from transmitting, and hear the packet be correctly retransmitted by the nodes at the next stage. Instead, only a few relays will suffice to correctly convey a packet to D . In this respect choosing, e.g., $\hat{N}^{\text{res}} = 2$ would achieve very good success probability with about half the transmissions compared to the case $\hat{N}^{\text{res}} = 4$. The ranking among the curves is progressively inverted when p increases, as in this case more relays are needed to guarantee success. For example, for $p = 0.5$, $\hat{N}^{\text{res}} = 4$ yields fewer transmissions than even $\hat{N}^{\text{res}} = 1$ (which is too restrictive, and often causes several silenced nodes to transmit, after they fail to hear the relays at the next stage forward the packets further on). Fig. 7 also suggests that $\hat{N}^{\text{res}} = 2$ and $\hat{N}^{\text{res}} = 3$ are good choices for almost all values of p , except perhaps if $p < 0.2$. Finally, note that the AR policy performs similar to unrestricted flooding in terms of probability of successful decoding at D (Fig. 5), but requires much fewer transmissions to attain this, as confirmed by Fig. 7.

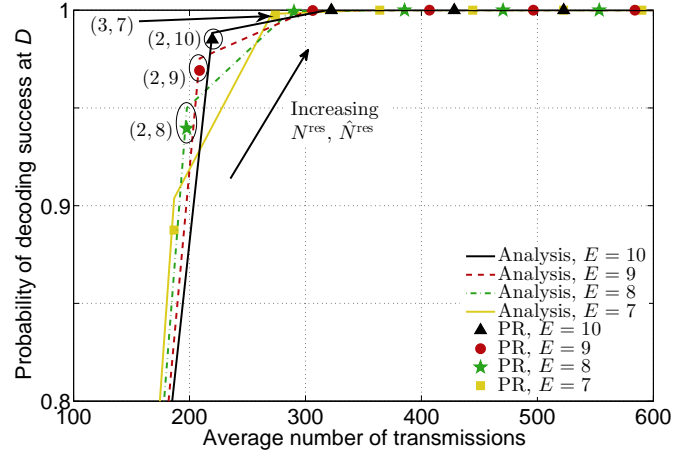


Figure 8. Interplay between the probability of decoding success at D and the average number of transmissions achieved by the PR policy, compared to restricted flooding, for $p = 0.5$ and for different values of N^{res} , \hat{N}^{res} , and E . Curves are spanned from left to right by increasing N^{res} (for restricted flooding) or \hat{N}^{res} (for the PR policy).

D. Insight from the restricted flooding model

The restricted flooding model described in Section III-C offers a good means to understand the behavior of the PR and AR policy. In Fig. 8, we consider the interplay between the probability of decoding success at D and the number of transmissions performed in the whole network. The graph shows a comparison between restricted flooding (lines) and the PR policy (markers), for $p = 0.5$ and for different values of N^{res} , \hat{N}^{res} and E . The restricted flooding curves are spanned from left to right by increasing N^{res} , the PR curves by increasing \hat{N}^{res} : in both cases the effect is to increase the probability of decoding success at the price of an increase in the number of transmissions. The figure marks four choices of the $(\hat{N}^{\text{res}}, E)$ pairs that achieve a probability of success of at least 0.9. We observe that the PR policy is an effective distributed implementation of restricted flooding, and that it helps achieve the same performance in terms of number of transmissions, at the price of a very small decrease in the success probability, mainly due to the distributed implementation. In any event, the mismatch becomes negligible for $\hat{N}^{\text{res}} \geq 3$.

Fig. 9 compares the performance of the AR policy against the analysis of restricted flooding for $p = 0.5$ and for different values of N^{res} , \hat{N}^{res} and E . The plot suggests that the AR policy behavior typically leads to an increased number of transmissions. However, this is compensated by the advantage of a higher probability of success. This is especially the case for low values of E , which implies that the typical overhead achieved by the AR policy is lower (see also Fig. 2). Once the probability of success has achieved a value of about 1, a further increase of \hat{N}^{res} yields a negligible improvement and, as expected, causes the number of transmissions to increase. In any event, the increase is limited with respect to plain restricted flooding, as the AR policy achieves a very high probability of success already for low values of E . For example, for $E = 2$ the AR policy already attains a probability of success close to 1, and requires fewer transmissions than plain restricted flooding in order to achieve the same result.

$$P_{\text{fail}}^{\text{hlp}}(q_r, q_v) \leq \sum_{j=0}^m \sum_{k=0}^{k_{\text{max}}} \binom{M}{j} q_r^{M-j} (1-q_r)^j \binom{E_h}{k} q_v^{E_h-k} (1-q_v)^k + \sum_{j=j_{\text{min}}}^M \sum_{k=k_{\text{min}}}^{E_h} \binom{M}{j} q_r^{M-j} (1-q_r)^j \binom{E_h}{k} q_v^{E_h-k} (1-q_v)^k \eta^{-j+k-m} \quad (28)$$

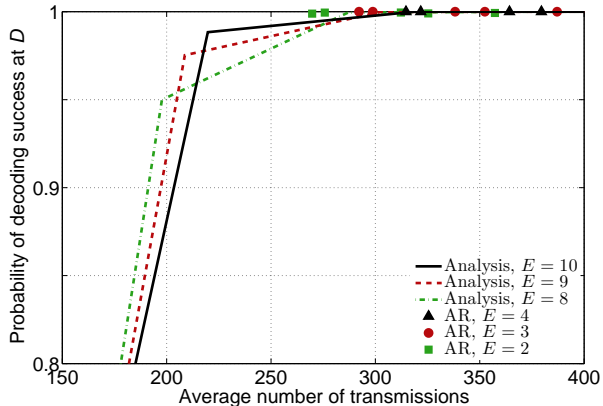


Figure 9. Interplay between the probability of decoding success at D and the average number of transmissions achieved by the AR policy, compared to restricted flooding, for $p = 0.5$ and for different values of N^{res} , \bar{N}^{res} , and E . Curves are spanned from left to right by increasing N^{res} (for restricted flooding).

V. EXTENSION: HELPER NODES

The study presented in this paper considers fountain-coded data flooded through a network up to a given destination, and provides fundamental insights on the interplay between the probability of success and the number of transmissions required to achieve it. Two policies are proposed to optimize the flooding process by reducing the total number of transmissions without decreasing the probability of success. The study lends itself to several extensions, in terms of both functionality and modeling. In the following subsection, we extend the study by allowing relays that successfully decoded the fountain-coded packets from the source to become “helper nodes,” and inject additional redundancy. In general, this redundancy will be different from that sent by the source, and thus it will help the nodes downstream decode the source packets.

A. Additional redundancy from intermediate “helper” nodes

In the following, we assume that there exists one node that received enough packets to decode the fountain code and reconstruct the original information sent by the source S . This node becomes able to inject additional redundancy packets. We call this node a helper node, and refer to it via the subscript h . We will now elaborate on the performance of the network in the presence of a helper, provide design guidelines for where this node should be placed, and discuss the utility yielded by the presence of additional helper nodes.

We start by noting that node h must belong to one of the relay stages between the source S and the destination D . In particular, we assume that out of the r stages that separate S from D , there are u stages between S and h , and v stages between h and D , so that $r = u + v + 1$. After successful

decoding, h transmits E_h additional redundancy packets, in order to increase the probability of successful decoding at D . For this analysis, we assume that only one helper node exists in the network.

We start from the probabilities q_u that a packet fails to reach node h after u relay stages, and q_v that D fails to receive an extra redundancy packet transmitted by h ; using the same formulation of (9), we can write

$$q_u = \mathbf{p}_S^T \left(\prod_{i=1}^u \mathbf{P}_i \right) \mathbf{p}_D, \quad q_v = \mathbf{p}_S^T \left(\prod_{i=1}^v \mathbf{P}_i \right) \mathbf{p}_D. \quad (25)$$

The probability of decoding failure at the relay stage of the helper node is found via q_u as

$$P_{\text{fail}}(q_u) \leq \sum_{j=0}^m \binom{M}{j} q_u^{M-j} (1-q_u)^j + \sum_{j=m+1}^M \binom{M}{j} q_u^{M-j} (1-q_u)^j \eta^{-(j-m)} \quad (26)$$

The probability that at least one node successfully decodes the fountain code (out of the N nodes belonging to the relay stage of node h) is $1 - P_{\text{fail}}(q_u)^N$. Now, a fountain decoding failure occurs at D in one of the following two cases: *i*) h fails to decode the fountain code and thus cannot help: in this case, D may fail to decode after the M packets transmitted by S according to the same arguments in Section III-A; *ii*) h successfully decodes the fountain-coded packets received from S and floods E_h extra redundancy packets towards D , but D still fails to decode. The total probability of failure is finally upper-bounded by

$$(1 - P_{\text{fail}}(q_u)^{N_{u+1}}) P_{\text{fail}}^{\text{hlp}}(q_r, q_v) + P_{\text{fail}}(q_u)^{N_{u+1}} P_{\text{fail}}(q_r), \quad (27)$$

where $P_{\text{fail}}^{\text{hlp}}(q_r, q_v)$ is the probability of fountain decoding failure at D , given the probability q_r that D fails the reception of a packet from S , and the probability q_v that D fails a reception from h . As in (6), we distinguish between two cases, namely that D receives up to m packets and that D receives more than m packets. In both cases, the packets may either come only from S or be a combination of packets from S and h . We have the formula in Eq. (28), where $k_{\text{max}} = \min\{E_h, m - j\}$, $j_{\text{min}} = \max\{0, m + 1 - E_h\}$ and $k_{\text{min}} = \max\{0, m + 1 - j\}$. In the computation of the average number of transmissions $T_{\text{avg}}^{\text{hlp}}$ in the presence of h , we need to account for the extra effort paid by the network to forward h 's packets. We have

$$T_{\text{avg}}^{\text{hlp}} = M T_{\text{avg}}(S \rightarrow D) + E_h T_{\text{avg}}(h \rightarrow D), \quad (29)$$

where $T_{\text{avg}}(S \rightarrow D) = 1 + \sum_{i=1}^r \mathbf{p}_S^T \left(\prod_{j=0}^{i-1} \mathbf{P}_j \right) \mathbf{t}_i$ and $T_{\text{avg}}(h \rightarrow D) = 1 + \sum_{i=1}^v \mathbf{p}_S^T \left(\prod_{j=0}^{i-1} \mathbf{P}_j \right) \mathbf{t}_i$ respectively

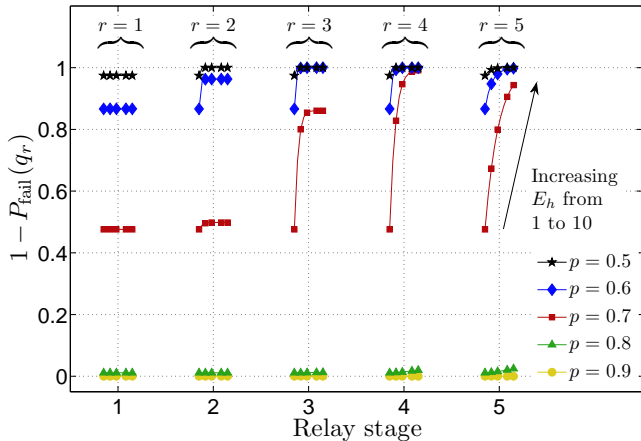


Figure 10. Analysis and simulation of $1 - P_{\text{fail}}(q_r)$ vs relay stage in the case $N_i = N = 10 \forall i$, for $m = 10$, $r = 5$, and $E = 1$, for varying number of redundancy packets E_h from the extra transmitting node. For higher values of the link error probability p , a helper node located at stage $r = 4$ yields better probability of success, whereas for lower values of p , $r = 3$ is a better choice.

represent the average number of transmissions from S to D and the average number of transmissions from h to D .

Fig. 10 plots the probability of successful decoding at the destination in the presence of a helper node, as a function of the relay stage where the helper is located, and for different values of the link error probability p , where $N = 10$, $r = 5$ and $E = 1$. We observe that the actual value of the success probability increases with the number of redundancy packets sent by h , E_h , and that the best performance for each value of p depends on the relay stage where h is located.

For example, consider the case of $p = 0.7$. Choosing to have h at stage 4 would make the success probability achieve a value of ≈ 1 for $E_h = 8$ packets. However, this is not the case if the helper node h is placed at any other relay stage, especially at the 1st or 5th stage. Similarly, for $p = 0.6$ and, say, $E_h = 3$, having the helper node at stage 3 achieves a probability of success very close to 1, whereas a placement in any other stage would be less effective.

While the position of node h that maximizes the success probability depends also on N and E , the example above shows our point that h should not be located too close to S or D . The intuition behind the above result is that if h were too close to S , it would not benefit from the flooding of source packets and the higher decoding probability that results. Conversely, if h were too close to D , any sufficiently high probability of error would make the packet flooding process die out before reaching h . In turn, h would not be able to reconstruct the source information and inject new redundancy packets. Moreover, even if h could decode the message, extra redundancy packets would not benefit from the flooding process on the way to the destination.

Assuming that h is located at stage 4, in Fig. 11 we show the interplay between the probability of decoding success at D and the number of transmissions in the network in the presence of h . We consider restricted flooding (see Section III-C) and plot different curves, each for a different value of E . All

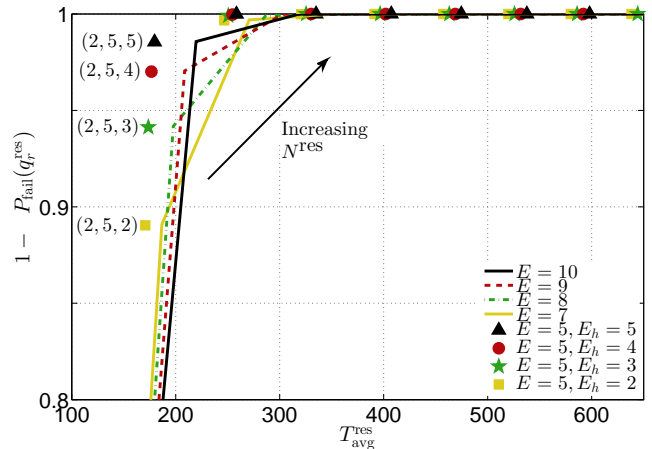


Figure 11. Interplay between $1 - P_{\text{fail}}(q_r^{\text{res}})$ and $T_{\text{avg}}^{\text{res}}$ for the case where only the source S transmits compared to the case with an extra helper node. For the former, we consider the analysis in Section III-C for different values of E ; for the latter, we show simulation results for different (N^{res}, E, E_h) triples, for $p = 0.5$. Curves are spanned from left to right by increasing N^{res} .

curves are spanned from left to right by increasing N^{res} . In addition to these curves, we plot a set of points described by the triplets (N^{res}, E, E_h) , which correspond to the cases where a helper node is present. We observe that the inclusion of h helps reduce the average number of transmissions (where the saving is larger for higher values of E). In fact, the best probability of success for the same total redundancy $E + E_h$ would be achieved if S sent all the redundancy itself. However, this would also lead to the largest possible number of transmissions because of the flooding process. Having a helper node between S and D helps save considerable resources by reducing the total transmissions while not harming the probability of success. This demonstrates the feasibility of the helper node solution.

We now consider the possibility of having either limited or unlimited helper nodes, and to allow each helper to inject either a limited or an unlimited number of redundancy packets.

More specifically, in the same simulation scenario considered so far, we allow any intermediate node that correctly received a sufficient number of packets from S to become a *potential* helper. When a helper injects additional redundancy packets, these may be received by intermediate nodes and thereby generate additional helpers in a sort of avalanche effect. Helpers act only if D failed to decode the fountain code via the source packets relayed by the network. In this case, we pick a helper node at random and let it transmit one redundancy packet each time. This packet will be flooded through the network between the helper and D and, if received by D , it will help decoding the fountain coded packets of S . The duration of this process and the set of helper nodes are regulated by the following four policies:

- 1) **Limited number of helpers, limited redundancy per helper:** only nodes that decoded the fountain code using the packets received from S can become helpers (no avalanche). Only one redundancy packet can be transmitted by each helper.
- 2) **Unlimited number of helpers, limited redundancy per**

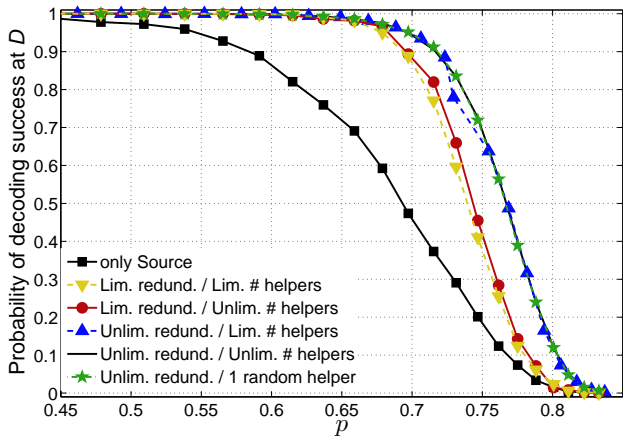


Figure 12. Probability of decoding success at D against p for different helper policies. The best performance is achieved when one or more helper nodes are allowed to send an unlimited number of redundancy packets. $N^{\text{res}} = 6$, $E = 2$.

helper: any node that decoded the fountain code using any set of packets, including those sent by other helpers, can become a helper itself (avalanche allowed). Only one redundancy packet can be transmitted by each helper.

3) **Limited number of helpers, Unlimited redundancy per helper**: helper avalanche not allowed, helpers can transmit an unlimited⁴ number of redundancy packets, until the destination decodes the packet successfully.

4) **Unlimited number of helpers, Unlimited redundancy per helper**: helper avalanche allowed, helpers can transmit an unlimited number of redundancy packets.

We compare the four policies above with the case of a single helper node chosen at random in the network, and allowed to transmit unlimited redundancy packets, akin to the case discussed for Figs. 10 and 11.

Fig. 12 shows the probability of successful decoding at D for $N^{\text{res}} = 6$ and $E = 2$. The leftmost curve corresponds to the case where only S transmits. Any helper node policy provides better performance. Specifically, the policies allowing only limited redundancy per helper start failing more often for $p \geq 0.7$, as transmission failures prevent D from receiving the helper packets; moreover, it becomes increasingly less likely that there are any helpers at all. Conversely, the policies that allow unlimited redundancy per helper achieve a very good performance. In particular, the policy allowing unlimited helpers and redundancy per helper only fails in those cases where no helper is present in the network.

The cost of the improvement in the probability of success is measured by the number of transmissions carried out to relay each packet in the network. For those cases where the helper nodes actually transmit, this metric is shown in Fig. 13. The policies with limited redundancy per helper progressively decrease their number of transmissions for $p \geq 0.7$, supporting the discussion above. Conversely, the policies allowing unlim-

⁴Allowing an unlimited number of redundancy packet transmissions per helper implies the design of a stopping rule or protocol, so that the destination can communicate that the fountain code has been correctly decoded. The design of such a rule is beyond the scope of this paper.

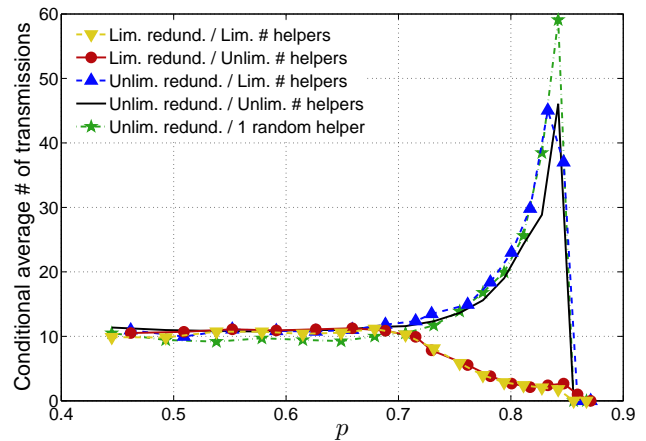


Figure 13. Conditional average number of transmissions per packet against p , for the only cases where helper nodes are needed to achieve correct decoding at D . The most aggressive helper policies experience failure only if p is sufficiently high to avoid that any helpers actually exist. $N^{\text{res}} = 6$, $E = 2$.

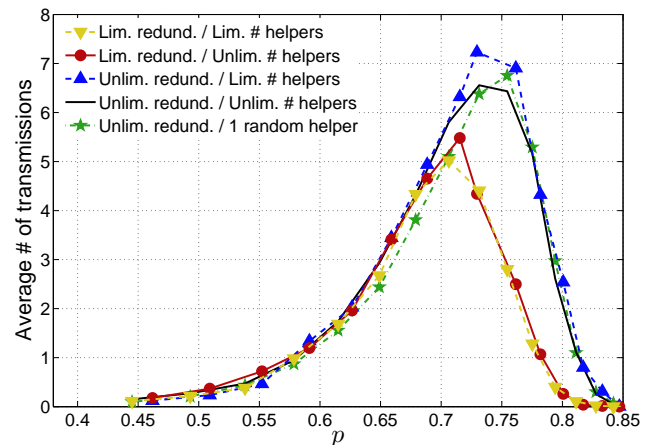


Figure 14. Average total number of transmissions per packet against p , including both the cases where helpers node are required and the cases where they are not. $N^{\text{res}} = 6$, $E = 2$.

ited redundancy keep increasing the number of transmissions until p is sufficiently high to prevent the generation of helpers in the network. The total average number of transmissions in Fig. 14 (taken over both the cases where helpers act and those where they do not), further confirms the above discussion, and shows that the largest number of transmissions is reached for very high values of p , as expected.

We remark that a helper node injects redundancy packets only when it has successfully decoded the source packets. Removing the latter constraint would mean that the injected redundancy could only help recover the set of source packets originally encoded into the fountain-coded packets received by the helper. The evaluation of this case is left as a future extension.

VI. CONCLUSIONS

We considered the flooding of fountain-coded packets through a multihop network, and analytically determined the

probability that the destination can recover the original data packets as a function of the number of nodes in the network, of the link error probability and of the amount of redundancy generated by the code. We showed that our model matches simulation results very well. Based on the conclusions drawn from the model, we designed practical, distributed policies that achieve the same success probability performance while requiring fewer (re)transmissions to advance a packet through the network. We measured the performance of these policies and justified their behavior in light of an analytical model for flooding under a restriction on the number of relays per hop.

Among several possible extensions to this work, we considered the case of intermediate relays being able to decode the source packets and thereby inject additional redundancy in the network. We thoroughly developed and evaluated this case. Future extensions may involve, e.g., the inclusion of plain retransmissions (without decoding the fountain code) or the explicit modeling of MAC-layer issues such as backoff mechanisms, interference and collisions.

REFERENCES

- [1] D. J. C. MacKay, "Fountain Codes," *IEE Proceedings – Communications*, vol. 152, no. 6, pp. 1062–1068, Dec. 2005.
- [2] M. Mitzenmacher, "Digital fountains: a survey and look forward," in *Proc. IEEE ITW*, San Antonio, TX, Oct. 2004.
- [3] M. Rossi, N. Bui, G. Zanca, L. Stabellini, R. Crepaldi, and M. Zorzi, "SYNAPSE++: Code dissemination in wireless sensor networks using fountain codes," *IEEE Trans. Mobile Comput.*, vol. 9, no. 12, pp. 1749–1765, Dec. 2010.
- [4] Z. Zhou, H. Mo, Y. Zhu, Z. Peng, J. Huang, and J.-H. Cui, "Fountain code based adaptive multi-hop reliable data transfer for underwater acoustic networks," in *Proc. IEEE ICC*, Ottawa, Canada, Jun. 2012.
- [5] A. James, A. S. Madhukumar, E. Kurniawan, and F. Adachi, "Performance analysis of fountain codes in multihop relay networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4379–4391, Nov. 2013.
- [6] R. Budde, S. Nowak, and R. Kays, "Reliable broadcast transmission in vehicular networks based on fountain codes," in *Proc. IEEE VTC Spring*, Budapest, Hungary, May 2011.
- [7] N. Kadi and K. Al Agha, "MPR-based flooding with distributed fountain network coding," in *Proc. IFIP Med-Hoc-Net*, Juan-les-pins, France, Jun. 2010.
- [8] A. Apavatjirut, K. E. Jaffrés-Runser, C. Goursaud, and C. Lauradoux, "Overflow of fountain codes in multi-hop wireless sensor networks," in *Proc. IEEE PIMRC*, Toronto, Canada, Sep. 2011.
- [9] E. Altman and F. de Pellegrini, "Forward correction and fountain codes in delay-tolerant networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 1, pp. 1–13, Feb. 2011.
- [10] M. Gerami and M. Xiao, "Repair for distributed storage systems with erasure channels," in *Proc. IEEE ICC*, Budapest, Hungary, Jun. 2013.
- [11] P. Casari, W. bin Abbas, and M. Zorzi, "On the number of transmissions vs. redundancy tradeoff for flooded fountain codes," in *Proc. IEEE CAMAD*, Athens, Greece, Dec. 2014.
- [12] A. James and A. Madhukumar, "Performance limits of rateless codes in delay constrained multihop relay networks," in *Proc. IEEE ICC*, Sydney, Australia, Jun. 2014.
- [13] A. Crismani, U. Schilcher, S. Toumpis, G. Brandner, and C. Bettstetter, "Packet travel times in wireless relay chains under spatially and temporally dependent interference," in *Proc. IEEE ICC*, Sydney, Australia, Jun. 2014.
- [14] G. Sim, J. Widmer, and B. Rengarajan, "Opportunistic finite horizon multicasting of erasure-coded data," *IEEE Trans. Mobile Comput.*, vol. 15, no. 3, pp. 705–718, Mar. 2016.
- [15] P. Casari, M. Rossi, and M. Zorzi, "Towards optimal broadcasting policies for HARQ based on fountain codes in underwater networks," in *Proc. IEEE/IFIP WONS*, Garmisch-Partenkirchen, Germany, Jan. 2008.
- [16] R. Diamant and L. Lampe, "Adaptive error-correction coding scheme for underwater acoustic communication networks," *IEEE J. Ocean. Eng.*, vol. 40, no. 1, pp. 104–114, Jan. 2014.
- [17] R. Ahmed, M. Stojanovic, and M. Chitre, "Random linear packet coding for broadcast networks," in *Proc. MTS/IEEE OCEANS*, St. John's, NL, Canada, Sep. 2014.
- [18] A. James and A. Madhukumar, "Enhanced packet delivery in delay limited rateless coded multihop networks," in *Proc. IEEE ICCS*, Macau, China, Nov. 2014.
- [19] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Proc. IEEE WCNC*, New Orleans, LA, Mar. 2003.
- [20] S. Crisostomo, U. Schilcher, C. Bettstetter, and J. Barros, "Analysis of probabilistic flooding: How do we choose the right coin?" in *Proc. IEEE ICC*, Dresden, Germany, Jun. 2009.
- [21] H. AlHazza, "Enhancing dynamic probabilistic broadcasting flooding scheme in MANETs," in *Proc. IEEE CHUSER*, Penang, Malaysia, Dec. 2011.
- [22] A. Hanashi, I. Awan, and M. Woodward, "Performance evaluation based on simulation of improving dynamic probabilistic flooding in MANETs," in *Proc. WAINA*, Bradford, UK, May 2009.
- [23] C. Betoule, T. Bonald, R. Clavier, D. Rossi, G. Rossini, and G. Thouenon, "Adaptive probabilistic flooding for multipath routing," in *Proc. NTMS*, Istanbul, Turkey, May 2012.
- [24] Y. Mylonas, M. Lestas, A. Pitsillides, P. Ioannou, and V. Papadopoulou, "Speed adaptive probabilistic flooding for vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 1973–1990, May 2015.
- [25] X. Shen, Y. Chen, Y. Zhang, J. Zhang, Q. Ge, G. Dai, and T. He, "OppCode: Correlated opportunistic coding for energy-efficient flooding in wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1631–1642, Dec. 2015.
- [26] T. Kunz, S. Paul, and L. Li, "Efficient broadcasting in tactical networks: Forwarding vs. network coding," in *Proc. IEEE MILCOM*, San Jose, CA, Oct. 2010.
- [27] H. Xi, X. Wang, Y. Zhao, and H. Zhang, "A reliable broadcast transmission approach based on random linear network coding," in *Proc. IEEE VTC Spring*, Yokohama, Japan, May 2012.
- [28] B. Vellambi, N. Rahnavard, and F. Fekri, "FTS: A distributed energy-efficient broadcasting scheme using fountain codes for multihop wireless networks," *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3561–3572, Dec. 2010.
- [29] G. Liva, E. Paolini, and M. Chiani, "Performance versus overhead for fountain codes over \mathbb{F}_q ," *IEEE Commun. Lett.*, vol. 14, no. 2, pp. 178–180, Feb. 2010.
- [30] H. M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, 3rd ed. Academic Press, 1998.



Waqas Bin Abbas received the Bachelors and the Masters degree from National University of Computer and Emerging Sciences (NUCES), Islamabad, Pakistan in 2008 and 2012, respectively, and the Ph.D. in Information Engineering in 2017 from the University of Padova, Italy. Currently, he is working as an Assistant Professor at NUCES, Islamabad, Pakistan. During Masters, his research was focused in underwater wireless communication, while during Ph.D., his research was mostly focused on energy efficiency in wireless networks. His current research interests include energy efficiency in 5G millimeter wave cellular networks, MIMO communication and multi-hop wireless networks.



Paolo Casari received the PhD in Information Engineering in 2008 from the University of Padova, Italy. After being on leave at the Massachusetts Institute of Technology in 2007, his research progressively focused on underwater communications and networks. He collaborated to several funded projects including CLAM (FP7), RACUN (EDA), and several US ARO, ONR and NSF initiatives. He was Technical Manager of the NAUTILUS and WISE-WAI projects and is now the scientific coordinator of the EU H2020 RECAP project. In 2015, he joined

the IMDEA Networks Institute, Madrid, Spain, where he leads the Ubiquitous Wireless Networks group. He served in the organizing committee of several conferences; he has been guest editor of a special issue of the Hindawi Journal of Electrical and Computer Engineering on “Underwater Communications and Networking,” and is currently co-guest editor of an IEEE Access special issue in the same field. His research interests include many aspects of underwater communications, such as channel modeling and simulation, network protocol design and evaluation, localization, and field experiments.



Michele Zorzi [F'07] (zorzi@dei.unipd.it) received his Laurea and Ph.D. degrees in electrical engineering from the University of Padova in 1990 and 1994, respectively. During academic year 1992/1993 he was on leave at the University of California San Diego (UCSD). After being affiliated with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy, the Center for Wireless Communications at UCSD, and the University of Ferrara, in November 2003 he joined the faculty of the Information Engineering Department of the

University of Padova, where he is currently a professor. His present research interests include performance evaluation in mobile communications systems, random access in mobile radio networks, ad hoc and sensor networks and IoT, energy constrained communications protocols, 5G millimeter-wave cellular systems, and underwater communications and networking. He was Editor-in-Chief of IEEE Wireless Communications from 2003 to 2005, Editor-in-Chief of IEEE Transactions on Communications from 2008 to 2011, and is currently the founding Editor-in-Chief of IEEE Transactions on Cognitive Communications and Networking. He was Guest Editor for several Special Issues in IEEE Personal Communications, IEEE Wireless Communications, IEEE Network, and IEEE JSAC. He served as a Member-at-Large in the Board of Governors of the IEEE Communications Society from 2009 to 2011, and as its Director of Education from 2014 to 2015.