# CRBcast: A Reliable and Energy-Efficient Broadcast Scheme for Wireless Sensor Networks Using Rateless Codes

Nazanin Rahnavard, *Member, IEEE*, Badri N. Vellambi, *Student Member, IEEE*, and Faramarz Fekri, *Senior Member, IEEE*

***Abstract*—This paper introduces a novel two-phase broadcast scheme referred to as *collaborative rateless broadcast* (CRBcast). CRBcast is a scalable approach for reliable and energy-efficient broadcasting in a multihop wireless sensor networks that also addresses load balancing, while requiring no knowledge of network topology. CRBcast combines the energy-efficiency offered by probabilistic broadcasting (PBcast) with the reliability features offered by application-layer rateless coding. In the first phase of CRBcast, packets encoded using a rateless code are dispersed into the network based on PBcast. In the second phase, simple collaboration of neighboring nodes ensures that all nodes recover original data with a very high probability of success. Since the performance of CRBcast rests heavily on that of PBcast, first part of this paper analyzes both analytically and via simulations the probabilistic broadcasting scheme. We then study the effectiveness of CRBcast. We show that CRBcast provides both reliability and energy efficiency simultaneously. Simulation results indicate that CRBcast provides an energy savings of at least $72\%$ and $60\%$ in comparison with flooding and PBcast, respectively.**

***Index Terms*—Energy-efficient broadcast, wireless sensor networks, probabilistic broadcast, rateless coding, collaborative broadcast.**

## I. INTRODUCTION

EFFICIENT network-wide broadcasting is an important issue in wireless networks that attracted a lot of attention. The type of broadcast data could be bulk data such as software files or short data such as route discovery packets. Some important factors that influence the efficiency of a broadcasting scheme can be reliability (defined as the percentage of nodes in the network that are able to retrieve the data), energy efficiency, complexity, scalability, and latency. Based on the application, some factors might be more important than others. For example, for updating the software in all the nodes in the network, reliability is very important, while latency might have less importance. Broadcasting streaming media is a case where latency is of paramount importance. Energy is usually an important issue especially for battery-powered sensor networks.

Manuscript received November 1, 2007; revised February 21, 2008; accepted April 20, 2008. The associate editor coordinating the review of this paper and approving it for publication was D. Wu.

N. Rahnavard is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: nazanin.rahnavard@okstate.edu).

B. N. Vellambi is with the Institute for Telecommunications Research, University of South Australia, Mawson Lakes, SA 5095, Australia (e-mail: badri.vellambi@unisa.edu.au).

F. Fekri is with the School of Electrical and Computer Engineering, the Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: fekri@ece.gatech.edu).

This work was partly done when N. Rahnavard and B. Vellambi were graduate students at the Georgia Institute of Technology.

Digital Object Identifier 10.1109/T-WC.2008.071218

In this paper, we consider the case that a large amount of packets have to be broadcast in a multihop wireless network with our main concerns being *reliability* and *energy efficiency*. We are also interested in a broadcasting scheme that has low complexity of implementation and is distributed and practical. In this paper, we address this problem for the following network model.

### A. Network Model

For our problem, we consider the following setup. We assume a wireless network of $N$ static nodes with omnidirectional antennas and transmission range $r$ deployed uniformly at random in a field with area $A$. We model the network by a random geometric graph $G(N, r)$. We also assume that $r$ is large enough so that $G(N, r)$ is connected.

Furthermore, we initially consider lossless networks by assuming lossless channels and the existence of a medium access control (MAC) layer, which prevents collision of packets. That is to say, every packet sent by a node correctly reaches all its neighbors. We also apply a more specific MAC scheme for our simulations in Section IV-C. In this MAC scheme, when a node is transmitting, all of its neighbors up to two hops will be silent and will not transmit. This avoids interference and the hidden terminal problem [1]. We refer to this MAC as *two-hop blocking MAC*. A similar MAC was considered in [2].

As in [3], in our model, we consider only the energy spent for RF transmissions. Therefore, the energy consumption is proportional to the number of packet transmissions in the network.

We note that although lossless stationary networks were considered for our preliminary investigation, our proposed protocol is flexible for extensions to lossy and mobile networks as we discuss in Section IV-A.

### B. Related Work

The most straightforward way to perform reliable broadcasting is flooding [4]. In the flooding method, a node rebroadcasts a packet that it receives for the first time. In a connected and lossless network, reliability is guaranteed by flooding. Although flooding is very simple and scalable, it has the following disadvantage. Many redundant rebroadcasts occur especially in dense networks, which overconsume precious network resources like energy and bandwidth. This problem is known as the broadcast storm problem [5]. To alleviate this problem and decrease the number of rebroadcasts, several protocols have been suggested. In general, the problem of *reliable* and *energy-efficient* broadcasting in wireless networks has different solutions in the two following models.

- In the first model, the nodes have only relaying capability. In this case, reliable and energy-efficient broadcasting in a wireless network is equivalent to the problem of finding a *minimum-connected dominating-set* (MCDS) for the corresponding network graph, if we model the network as a geometric graph. Unfortunately, determining an MCDS is an NP-complete problem [6] even if a centralized algorithm utilizing the full knowledge of the graph topology is applied. This forces the employment of some heuristic and suboptimal schemes. One of the important schemes is called *probabilistic broadcast* (PBcast) [5], [7]. This approach was originally introduced in [8] to reduce traffic for multicast wired networks and later was tailored for wireless applications. In PBcast, a source broadcasts all the packets with probability one. Any other node rebroadcasts every packet that it receives for the first time with some probability $p < 1$. Therefore, the number of unnecessary rebroadcasts is decreased. However, some nodes may not receive all the packets. A high value for $p$ may be chosen to achieve reliability; however, if $p$ is too high, energy efficiency will be lost. Some other heuristic algorithms for attacking this problem have been proposed, e.g., [2], [3], [5], [7], [9], [10]. Most of them assume considerable knowledge of network topology and are either impractical or suffer from lack of reliability.
  Note that in all the above schemes, to ensure reliability one, every single packet needs to be received by all the nodes in the network. This constraint may cause lots of retransmissions and may require in-sequence data delivery. Moreover, these schemes would be far less efficient for lossy networks (i.e., networks with unreliable communication links).
- In the second model, in addition to relaying, each node has the capability of doing local processing and coding. This model was first introduced in [11] and opened a new research path known as *network coding*. Considerable work has been done in the area including [12], [13] and references therein. In this model, the problem is solvable by a polynomial-time algorithm, assuming that the network is directed. Network coding (NC)[†] can be divided into two decoupled problems. The first one is to determine the subgraph over which coding has to be performed and the flow rate on each link. The other is to determine the code to use over that subgraph. Linear programming (LP) aids in solving the first problem. The coding problem is solved by sending random linear combinations of received packets such that the coefficients of the linear combinations are selected from a finite field $GF(q)$, for which $q$ must be sufficiently large. We will refer to this type of coding as *random sum coding*. Clearly, information about the chosen coefficients must also be sent along with the sent packet for the decoding purpose. We note the following shortcomings with NC. First, the size of the finite field $GF(q)$ from which the coefficients of linear combinations are selected must be very large for optimality. This makes the computations costly. Second, the decoding is required to be Gaussian elimination with cubic complexity with respect to $n_p$, where $n_p$ is the number of original packets. Third, the overhead of random sum coding (due to transmission of coefficients with each packet) is $n_p \log_2 q$ bits for each sent packet. This overhead might be prohibitive if $n_p \log_2 q$ is comparable to the size of the packets. Forth, the assumption of directed graph is a limiting one, since wireless networks consisting of nodes with omnidirectional antennas are, by their nature, not directed, i.e., for any two nodes $i$ and $j$ that are in the transmission range of each other, the transmission can happen in both directions from $i$ to $j$ or from $j$ to $i$ in different time slots. The issue of finding optimal directions for the edges is an intractable problem by itself considering the fact that possible combinations of assignable directions grows exponentially with the number of edges in the network. Finally, application of NC (while ensuring optimality) to mobile networks may also be very difficult because the network topology is dynamic, and the wireless channel is noisy and time varying. Existing network coding schemes would require some predictive mechanism regarding the movement, traffic trends, and channel status. Implementation of network coding in dynamic networks with such information would require dynamic programming and may not be feasible.

Instead of using random sum codes that have cubic complexity of decoding, other coding techniques can also be employed at each node. One of the best options is *rateless erasure coding* [16]–[18]. Rateless codes have linear encoding and decoding complexity. The encoding is a low-weight packet-level addition of input packets over $GF(2)$, and the decoding is done by a simple iterative decoding. Unlike traditional codes, rateless codes do not assume any knowledge of the channel and are adaptable to different channel conditions. In [18]–[21], the applicability of rateless codes for reliable multicast/broadcast in *single-hop* lossy networks was mentioned. The original packets are first encoded using a rateless code. The encoded packets are then broadcast. In single-hop broadcasting using rateless codes, the redundancy is optimal for all clients independent of their packet loss rates. No prior knowledge of the channel status is needed. The client with higher loss rates has to wait longer to receive enough packets to recover the original data. However, the performance of broadcasting encoded data over *multihop* wireless networks depends on the underlying routing scheme as well. One option

[†]Although *network coding* is a broad term and can refer to any scheme in which the nodes in the network do more than just relaying, from now on we use this term to refer to network coding as defined in [14], [15].

is to find the optimal sub-network as in the case of NC (using LP) and then use rateless coding over the sub-network. However, the problem of finding routes using LP is not very practical for large networks such as sensor networks.

### C. Contribution of the paper

This paper proposes a two-phase reliable and energy-efficient broadcast scheme, referred to as *collaborative rateless broadcast* (CRBcast). CRBcast is a distributed and practical scheme that employs application-layer rateless coding in conjunction with a simple, and scalable routing protocol to guarantee reliability and increase energy efficiency of broadcasting in wireless sensor networks. In the first phase of CRBcast, the rateless-encoded packets are broadcast based on PBcast, in which each node probabilistically relays every new received packet. The second recovery phase, which is based on collaborations of the nodes, ensures that all the nodes can recover the original data. The collaboration of nodes is based on a simple advertisement and request mechanism inspired by the SPIN protocol [22].

As a result of broadcasting *rateless-encoded* packets, nodes in a network only require to receive enough number of packets rather than all broadcast data. However, to minimize the overhead incurred by the coding scheme, the number of original packets should not be too small.

We examine CRBcast analytically and by simulations. To do so, we also need to study PBcast since the characteristics of PBcast influence CRBcast. PBcast has been studied before by simulations in some studies, e.g., [7], [23], [24]. We elaborate the problem here and provide asymptotic analysis for finding the optimal forwarding probability. Preliminary results were initially introduced in [25].

The rest of the paper is organized as follows. Section II provides asymptotic analysis of PBcast. In Section III we introduce the idea of rateless encoding of data at source and then using PBcast to broadcast the data. This scheme is referred to as *rateless probabilistic broadcast* (RBcast). Simulation results are provided and indicates the gain that can be achieve by this approach. In Section IV, we propose and develop CRBcast and simulation results are provided. Finally, we conclude the paper in Section V.

## II. Asymptotic Analysis of Probabilistic Broadcast

PBcast is a scalable and simple scheme for broadcasting in multi-hop wireless networks. In PBcast, every node relays a packet that it receives for the first time with some probability $p$. Let us assume that at any time slot, in which a packet travels in the network, we color each node as black or white with probability $p$ and $1-p$, respectively. Therefore, black nodes forward a new received packet while white nodes do not forward it. Suppose $B$ and $W$ are the sets of the black and white nodes, respectively. Let $G_B(p,r) = G(N,r)\backslash W$ represents the subgraph of $G(N,r)$ induced by $B$. The following remark can be concluded.

*Remark:* The problem of energy-efficient and reliable broadcasting in wireless networks using PBcast can be rephrased as finding the lowest $p$ such that $G_B(p,r)$ is connected and every white node is in the single-hop neighborhood of at least one black node.

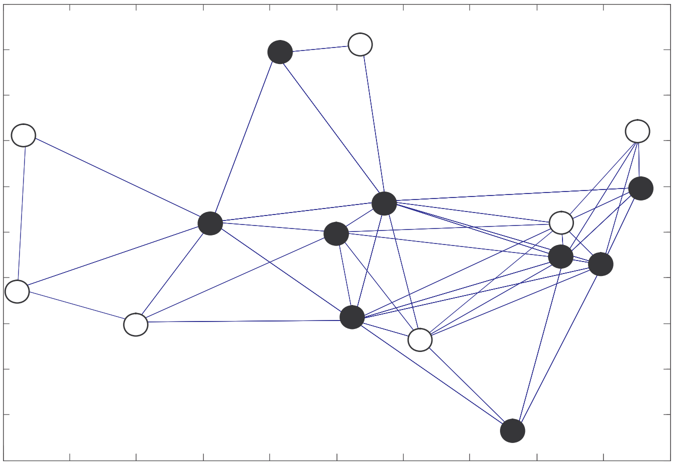

Fig. 1. A random deployment of 16 nodes in a field. At any instance, each node is colored black with probability $p$ and is colored white with probability $1-p$. In this figure $p = 1/2$.

For clarification, consider Figure 1 in which a random deployment of 16 nodes is depicted. The source node is in the center. The forwarding probability is $p = 1/2$. There is an edge between two nodes if they are in the transmission range of each other. All the black nodes are connected, and every white node has at least one black node as its neighbor. Therefore, if the source node broadcasts a packet, all the nodes in $G(N,r)$ receive it. This implies that reliability of PBcast is one in this case (although the number of transmissions is not minimum). Next, we will show that there exists a threshold $p^{th}$ such that the reliability of PBcast is equal to one asymptotically almost surely if and only if $p > p^{th}$.

### A. Connectivity of Black Nodes

Our goal is to find the condition under which $G_B(p,r)$ is connected. Connectivity of geometric graphs is a well studied subject. Gupta *et al.* [26] derived the condition for asymptotic connectivity of a random geometric graph. Later, [27] generalized the case and considered the connectivity of a random geometric graph with unreliable nodes in which each node fails with probability $1-p$. Interestingly, the connectivity of random graphs with unreliable nodes can be used to find the connectivity condition in $G_B(p(N), r(N))$ by mapping the failed nodes into white nodes and the active nodes into black nodes. The following theorem states the necessary and sufficient condition for connectivity of $G_B(p(N), r(N))$ for large values of $N$. For proof we refer readers to [27].

*Theorem 1:* Consider a random graph $G_B(p(N), r(N))$. Let $A$ denote the area of a square field in which we deployed the $N$ nodes at random. Assume $p(N) \times N \to \infty$ as $N \to \infty$ and let $\omega(N)$ be any slowly growing function such that $\omega(N) \to \infty$ as $N \to \infty$. Suppose we have

$$\lim_{N\to\infty} \left( \frac{p(N) N \pi r^2(N)}{(\ln(p(N)N) + \omega(N))\, A} \right) = \alpha. \qquad (1)$$

If $\alpha > 1$, then $G_B(p(N), r(N))$ is connected asymptotically almost surely. On the other hand, if $\alpha < 1$, then

$G_B(p(N), r(N))$ is not connected asymptotically almost surely.

Using Theorem 1, we conclude that $p^{th}$ for the connectivity of $G_B(p(N), r(N))$ is given by:

$$\frac{p^{th}\pi r^2(N)}{A} = \frac{\ln(p^{th}N) + \omega(N)}{N} \quad \text{as } N \to \infty. \tag{2}$$

### B. Sufficient Condition for the Coverage of White Nodes

In this section, we examine the sufficient condition under which every white node has at least one black node as its neighbor. We prove that (2) is a sufficient condition.

*Theorem 2:* $p > p^{th}$, where $p^{th}$ satisfies (2), provides a sufficient condition for the coverage of all white nodes.

*Proof:* Let us define $C_i$ as the event that the $i^{th}$ node has at least one black node neighbor (within a single hop) provided that the $i^{th}$ node is white. We show that asymptotically $Pr(\bigcap_{i=1}^{N} C_i)$ goes to one if (2) is satisfied. By union bound we have:

$$Pr(\bigcap_{i=1}^{N} C_i) \geq 1 - NPr(\overline{C_i}). \tag{3}$$

Moreover,

$$\begin{aligned} Pr(\overline{C_i}) &= \left(p(1 - \frac{\pi r^2}{A}) + (1-p)\right)^{N-1} \\ &= \left(1 - p\frac{\pi r^2}{A}\right)^{N-1}. \end{aligned} \tag{4}$$

Using inequality $1 - x < e^{-x}$ and (2) we have:

$$\begin{aligned} \left(1 - p\frac{\pi r^2}{A}\right)^{N-1} &< e^{-p\frac{\pi r^2}{A}(N-1)} \\ &< e^{-\ln(p^{th}N) - \omega(N)}. \end{aligned} \tag{5}$$

Therefore,

$$Pr(\bigcap_{i=1}^{N} C_i) \geq 1 - \frac{N}{p^{th}N} e^{-\omega(N)} \to 1 \quad \text{as } N \to \infty. \tag{6}$$

This concludes the assertion. ■

From Theorems 1 and 2, we conclude the following corollary.

*Corollary 1:* Broadcasting a single packet to a network by PBcast achieves reliability one asymptotically almost surely if and only if $p > p^{th}$, where $p^{th}$ satisfies (2).

So far, we have considered the case of broadcasting *a single packet*. We now study the case of broadcasting $n_p$ packets in a multi-hop wireless network. Let $R_1$ denote the fraction of nodes that receive a particular packet in PBcast. It is clear that in a uniform packet dissemination, $R_1$ also denotes the probability that a node receives the packet. Since the transmissions of packets are independent, the probability $R_{n_p}$ that a node receives all $n_p$ packets is equal to $R_1^{n_p}$ for a uniform packet dissemination. However, packet transmissions are not uniform in general. For instance nodes at the neighborhood of the source always receive the packets while the border nodes receive less packets. Therefore, $R_{n_p}$ is not equal to $R_1^{n_p}$ in general. In the next lemma, we derive bounds on $R_{n_p}$ †.

*Lemma 1:* Consider the PBcast protocol for broadcasting $n_p$ packets in a large wireless network with $N$ nodes. Let $R_1$ and $R_{n_p}$ denote the probabilities that a random node in the network receives a particular packet and $n_p$ packets, respectively. Then,

$$R_1^{n_p} \leq R_{n_p} \leq R_1. \tag{7}$$

*Proof:* The right-hand side inequality is obvious. We prove the left-hand side inequality. Let us partition the nodes in the network into groups $1, 2, \ldots, j$ such that the packet dissemination in each group is uniform. Let the fraction of the nodes in the $i^{th}$ group be $\alpha_i$ ($0 < \alpha_i \leq 1$ and $\sum_{i=1}^{j} \alpha_i = 1$). Also, let $R_{1,i}$ be the probability that a node in the $i^{th}$ group receives a packet. A circular partitioning may be suitable since the nodes with similar distances from the source are expected to have the same probability of receiving the packets. Assuming that the number of nodes in each partition is large enough, it is clear that we have

$$R_1 = \sum_{i=1}^{j} \alpha_i R_{1,i}. \tag{8}$$

Since the packet transmissions are independent in PBcast and assuming uniform reception of data in each group of nodes, we have

$$R_{np} = \sum_{i=1}^{j} \alpha_i R_{1,i}^{n_p}. \tag{9}$$

Using Jensen's inequality, we conclude that $R_{n_p} \geq R_1^{n_p}$. ■

Since energy consumption is proportional to the number of packet transmissions, it is also desirable to obtain the total number of required transmissions per original packet ($\frac{N_{tx}}{n_p}$). Since not all the black nodes receive a packet to transmit, $\frac{N_{tx}}{n_p}$ in PBcast is upper bounded by the total number of black nodes plus one (for the source node), which is equal to $pN + 1$ on the average. Moreover, in the area spanned by the nodes that receive a particular packet, on the average, fraction $p$ of them are transmitting nodes. Therefore, we have

$$\frac{N_{tx}}{n_p} = pNR_1. \tag{10}$$

### C. Simulation Results for Probabilistic Broadcast

In this section, we demonstrate PBcast properties by simulation. First, we consider random deployment of $N$ nodes uniformly in a field with area $A = 2000m \times 2000m$, for $N = 10^4, 10^5$, and $10^6$. For each $N$, we chose transmission range $r$ using $\frac{\pi r^2}{A} > \frac{\ln N}{N}$ [26], so that the network $G(N, r)$ is connected. We computed $p^{th}$; the threshold for the reliable broadcasting of a single packet from (2). Table I gives the analytical results. We note that as $N$ increases the required $p^{th}$ decreases.

†Alternatively, broadcasting $n_p$ packets can be performed such that a node keeps its forwarding status during the broadcasting session. This implies that a node is a forwarding node (with probability $p$) for the whole broadcasting session. In this case $R_{n_p} = R_1$. However, this unevenly distributes the energy consumption in the network, i.e., the fixed forwarding nodes consume much more energy than non-forwarding nodes. This is not desirable especially for sensor networks.

TABLE I
THE VALUES OF $p^{th}$ FOR RELIABLE BROADCASTING OF A SINGLE PACKET IN THE GEOMETRIC GRAPH $G(N, r)$ DEPLOYED RANDOMLY IN AN AREA $A = 2000m \times 2000m$.

| $N$ | $r(m)$ | $p^{th}$ |
|---|---|---|
| $10^4$ | 50 | 0.43 |
| $10^5$ | 20 | 0.34 |
| $10^6$ | 8 | 0.25 |

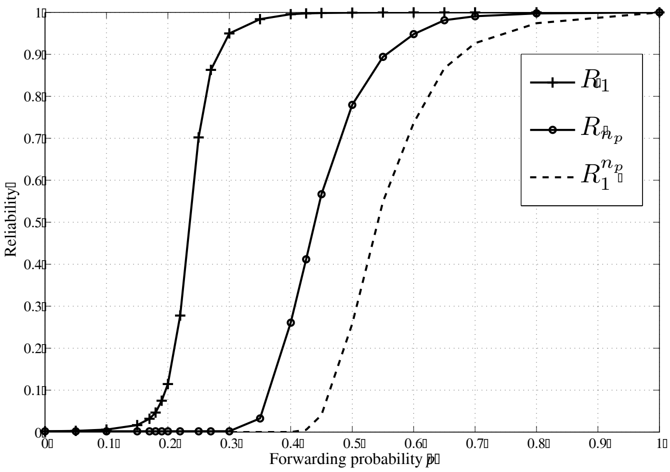


Fig. 2. $R_1$, $R_{n_p}$, and $R_1^{n_p}$ versus forwarding probability $p$ for a wireless sensor network with the topology $\mathcal{T}_1$ ($n_p$ is equal to 2000).

Next, we consider the following network topology. We assume $N = 10^4$ nodes with transmission range $r = 50m$ are deployed uniformly at random in an area $A = 2000m \times 2000m$. We call this topology as $\mathcal{T}_1$ for our future references. We developed event-driven softwares in C++ for our simulations.

We first verify our theoretical analysis for $p^{th}$. We also confirm that for reliable broadcasting of $n_p > 1$ packets, the required relaying probability is much higher than $p^{th}$. In Figure 2, the fraction of nodes that successfully receive a particular packet (denoted by $R_1$) and $n_p = 2000$ packets (denoted by $R_{n_p}$) are shown. Each point in the figure is the result of averaging over 300 different random graphs with the topology $\mathcal{T}_1$. We also depicted $R_1^{n_p}$, which is a lower bound for $R_{n_p}$ by Lemma 1. We confirm that $R_1$ is very close to one for $p > 0.43$, which is the analytical threshold value given by Table I. However, for $n_p > 1$, the reliability decreases and a larger forwarding probability $p$ is needed. We note that for $R_{n_p} \approx 1$ in PBcast, $p$ has to be very close to one. For example, forwarding probability of at least $p = 0.7$ is required for $R_{n_p} \geq 0.99$.

Next, we give the simulation results for the required number of transmissions per packet ($\frac{N_{tx}}{n_p}$) versus the forwarding probability $p$, since it is the criterion for energy consumption. Figure 3 shows $\frac{N_{tx}}{n_p}$ versus $p$ for the topology $\mathcal{T}_1$ when $n_p = 2000$.

We note that when the number of required transmissions per packet is plotted using Equation (10), we get the same result as the simulation provided in Figure 3. As we can see $\frac{N_{tx}}{n_p}$ is an increasing function of $p$. The greatest rate of increase in $\frac{N_{tx}}{n_p}$ happens around $p \approx 0.24$. This point is the threshold for

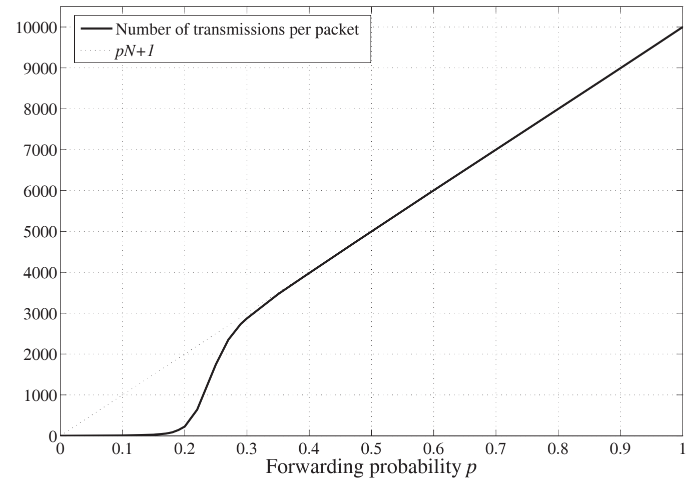


Fig. 3. $\frac{N_{tx}}{n_p}$ versus forwarding probability $p$ for the network topology $\mathcal{T}_1$.

occurrence of a *giant component* in $G_B(p, r)$. Let $p^G$ denote this threshold. As given in [27], the asymptotic value of $p^G$ can be calculated by

$$p^G N r^2 / A = \lambda_c \approx 1.44. \quad (11)$$

We refer to the giant component again in Section IV, where we observe that the optimal value of $p$ in our proposed protocol ($p^*$) is close to $p^G$. Therefore, (11) can be used for the approximation of $p^*$.

## III. RBCAST: RATELESS PROBABILISTIC BROADCAST

We noted in Section II-C that as the number of packets for broadcasting ($n_p$) increases, the forwarding probability $p$ also has to increase so that PBcast provides high reliability. The reason is that every single packet must be received by all the nodes in the network; however, as the number of packets increases the probability that a subset of nodes miss some packets increases as well. Therefore, as $n_p$ increases, the performance of a reliable PBcast becomes close to the performance of flooding ($p = 1$), and PBcast will not be energy efficient.

To overcome this problem, we investigate the following potential solution. The source node encodes the data using a channel code before broadcasting it. The encoded packets then are broadcast using PBcast. By doing this, nodes in the network do not require to receive all the broadcast packets. They only need to receive enough packets to be able to decode the data. We use *rateless codes*, because these codes do not require any information about the channel and also for their simple encoding and decoding. A node is able to decode and retrieve the original $n_p$ packets if it receives at least $n_p\gamma$ encoded packets, where $\gamma \geq 1$ is the overhead of rateless codes†. We refer to this scheme as *rateless probabilistic broadcast* (RBcast).

Next, we consider broadcasting $n_p = 2000$ packets over a random graph with the topology $\mathcal{T}_1$ using RBcast. In our simulations, we use the rateless code in [16] that results in

†The reader may refer to Appendix A and [16]–[18] for more details on encoding, decoding, and design of rateless codes.

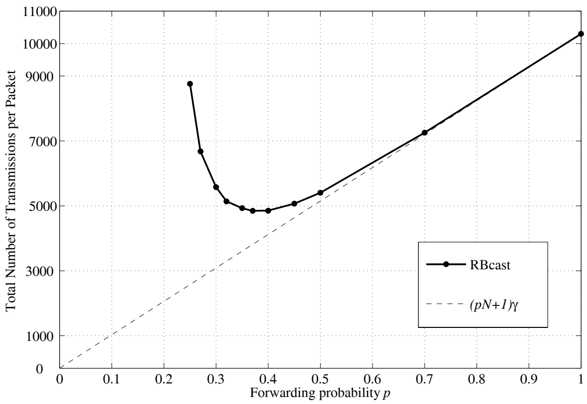


Fig. 4. The number of transmissions per packet versus the forwarding probability $p$ in RBcast. We considered $n_p = 2000$ and $\gamma = 1.03$ for a random network with topology $\mathcal{T}_1$.

a decoding failure probability less than $10^{-8}$ for $n_p = 2000$ and $\gamma = 1.03$. The source node generates encoded packets, and these packets are broadcast in the network based on PBcast. The broadcast session ends if all the nodes in the network receive at least $n_p\gamma = 2060$ packets. In this way, it is guaranteed that all the nodes can retrieve the original data with a probability very close to unity. Figure 4 depicts the number of transmissions per packet as a function of $p$ for RBcast.

As is shown, when $p$ is very small, a large number of transmissions is required. The minimum number of transmissions per packet is about 4850 and happens at $p = 0.37 \sim 0.4$. To compare RBcast with PBcast, we recall that $\frac{N_{tx}}{n_p} \approx 7000$ transmissions are necessary for gaining reliability of 99% in PBcast. Therefore, RBcast results in 30% less transmissions in comparison with PBcast. This result is promising; however, we next propose another broadcasting scheme that is similar to RBcast and improves the efficiency even more.

## IV. CRBcast: Collaborative Rateless Broadcast

In this section, we propose a scheme for reliable and energy-efficient broadcasting in multihop wireless networks. In the proposed scheme, we provide reliability by employing rateless coding. In rateless coding, potentially unlimited number of encoded packets can be generated by simple XOR operations on the original packets. A receiver is able to retrieve the original packets when it receives a sufficient subset of the encoded packets. No information about the channel is needed, and there is no need for in-sequence data delivery. In other words, it is not important as to which encoded packets a node receives. The only important thing is the number of the received encoded packets. We propose to use rateless codes in conjunction with a light-weight PBcast algorithm, which is a simple and scalable broadcasting scheme. In light-weight PBcast, we choose a small value for $p$. Light-weight PBcast reduces the probability of multiple reception of the same packet. This prevents many redundant transmissions. Therefore, the total number of transmissions decreases as does energy consumption. Since not every node in the network can recover the original packets, we need a second recovery phase to guarantee reliability. Next, we explain our proposed scheme, which we call *collaborative rateless broadcast* (CRBcast).

### A. The CRBcast Protocol

CRBcast consists of the following two phases.

*1) CRBcast-Phase I:* In Phase $I$, the original $n_p$ packets at source are first encoded to $n_p\gamma$ encoded packets. Then, the encoded packets are broadcast using PBcast.

At the end of Phase $I$, some nodes, referred to as *complete nodes*, receive all $n_p\gamma$ different packets and can reconstruct the original packets. We refer to the rest of the nodes that did not receive $n_p\gamma$ different packets as *incomplete nodes*. The number of complete nodes after Phase $I$ and the number of transmissions per packet can be approximated by $NR_{(n_p\gamma)}$ and $\frac{N_{tx}}{n_p} = pNR_1\gamma$, respectively, as discussed in Section II.

The parameter $\gamma \geq 1$ is the overhead imposed by the rateless coding and is selected such that the probability of successful decoding $P_R(n_p, \gamma)$ is almost one. For large values of $n_p$, the probability $P_R$ for rateless codes described in [16], [18] approaches one when $\gamma$ is slightly greater than one, while the complexity of encoding and decoding is linear.

*2) CRBcast-Phase II:* Phase $II$ is based on a simple collaboration among complete and incomplete nodes such that each complete node sends only once the required number of packets to its neighbors to complete them. The new complete nodes repeat this and the process continues until no new complete nodes remain. Therefore, we need two types of very short handshake messages between complete and incomplete nodes: advertisement messages (ADV) and request messages (REQ). Whenever a node becomes complete, it advertises its completeness to its neighbors once using an ADV message (which includes the ID of the complete node and some flag bits that indicate the message is an ADV message). Any incomplete neighbor that receives the ADV message responds by a REQ message including the required number of new packets for its completion, the ID of the complete node, and some flag bits associated with the REQ message. For example, suppose a node has already received $n_1$ packets when it receives an ADV message. This node would use an REQ message asking for $n_p\gamma - n_1$ new packets. It should be noted that the complete node generates new encoded packets by performing rateless encoding on the retrieved original packets. In this way, it can be guaranteed that newly encoded packets are sent to the incomplete node.

Here are more details about the collaboration between nodes. If an incomplete node receives multiple ADV messages from different neighbors, it will respond only to the one with the lowest ID number. Moreover, after sending an ADV message, each complete node waits long enough to receive the REQ messages from all of its incomplete neighbors. When a complete node receives different REQ messages from different incomplete neighbors, it sends the maximum number of the required packets. Each complete node advertises once and each incomplete node requests once. In a connected network with lossless channels, we prove in Lemma 2 that all nodes are eventually completed. The total number of ADV messages ($n_{adv}$) plus REQ messages ($n_{req}$) is less than $2N$ packets for the whole broadcasting session (equivalently, $2N/n_p$ transmissions per packet). These handshake messages result in negligible overhead due to their relatively short packet sizes and the fact that $n_p$ is large.

*Lemma 2:* Let $G(N, r)$ be a lossless, stationary, and connected wireless network. Then, CRBcast provides reliability one independent of the value $p$.

*Proof:* Let $S$ and $I$ denote the set of complete and incomplete nodes after the completion of Phase $II$, respectively. Clearly, $S$ and $I$ are disjoint and $|S| + |I| = N$. To prove that CRBcast provides reliability one, it is sufficient to prove that $I$ is an empty set. We note that $S$ is a nonempty set since the neighbors of the source are definitely complete. If $|S| = N$ then we are done. Otherwise, connectivity of $G(N, r)$ implies that there exists at least an edge $(i, j)$ such that $i \in S$ and $j \in I$. However, this contradicts the protocol in Phase $II$. Since, once node $i$ becomes complete, it completes all its incomplete neighbors based on the ADV and REQ mechanism in CRBcast. Therefore, $j$ is also complete, i.e., $j \in S$, which contradicts our assumption. ■

So far, we considered lossless stationary networks. Generalization of CRBcast for lossy networks is straightforward. The only required assumption is that each node needs to know its single-hop neighbors. With this assumption, CRBcast can simply be modified for lossy networks. Phase $I$ remains the same (however, a higher $p$ would be required). In Phase $II$, each complete node including the source node sends ADV. Lost ADV messages can be compensated by periodical re-advertisements. Lost REQ and data packets can be compensated by requesting the number of necessary packets that do not arrive within a fixed time period. When a node receives enough number of packets, the node declares this by sending REQ for *zero* packet in response to an ADV message. A complete node stops sending ADV when it receives REQ for zero packet from all its neighbors.

### B. Simulation Results for CRBcast: Time-Relaxed Implementation

Here, we provide the simulation results for the number of transmissions per packet ($\frac{N_{tx}}{n_p}$) that is necessary for all nodes to receive at least $n_p\gamma$ distinct encoded packets as a function of $p$. This quantity is the indication of the energy consumption in the network. In our simulations, we used the rateless code in [16] that results in a decoding failure probability less than $10^{-8}$ for $n_p = 2000$ and $\gamma = 1.03$. Therefore, the recovery probability ($P_R$) is almost one. We first considered the network topology $\mathcal{T}_1$ described in Section II. Figure 5 depicts the result. We also included the number of transmissions per packet in Phase $I$ and $II$. As expected, the total number of transmissions in Phase $I$ increases with $p$, while the total number of transmissions in Phase $II$ decreases with $p$. The minimum total number of transmissions occurs at $p^* = 0.25$, for which $\frac{N_{tx}}{n_p}$ is equal to 2769 transmissions per packet. This results in saving of more than 72% packet transmissions in comparison with flooding. To compare CRBcast with PBcast, we recall that $\frac{N_{tx}}{n_p} \approx 7000$ transmissions are necessary for gaining reliability of 99% in PBcast. Therefore, CRBcast results in 60% less transmissions in comparison with PBcast, while the reliability offered by CRBcast is almost one. To compare CRBcast with RBcast, we recall that RBcast can achieve the

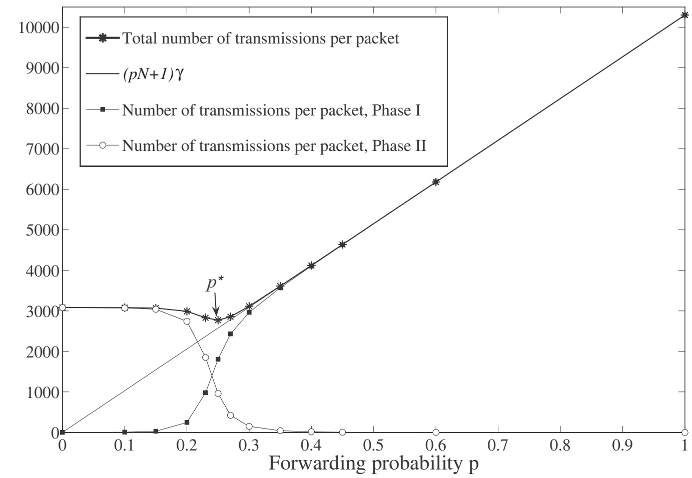


Fig. 5. The number of transmissions per packet versus the forwarding probability $p$ in CRBcast. We considered $n_p = 2000$ and $\gamma = 1.03$ for the network topology $\mathcal{T}_1$.

minimum of $\frac{N_{tx}}{n_p} \approx 4850$. Therefore, CRBcast outperforms RBcast in terms of number of transmissions per packet by about 43%. This shows the effectiveness of introducing the second recovery phase in CRBcast.

It is worth noting that using (11), the giant component for $G_B(p, r)$ happens theoretically at $p^G \approx 0.24$. We observe that $p^*$ is close to $p^G$. We can explain this by recalling that at $p^G$ a large fraction of nodes receive each packet (in Phase $I$). This balances the recoveries in the two phases. Therefore, (11) can be used as an approximation for $p^*$. We should also mention that the number of handshaking transmissions is at most $2N = 20000$ packets, which is about 0.3% of the total number of transmissions. Considering that handshake packets are much shorter than data packets, we see that this overhead is wholly negligible.

Table II summarizes $\frac{N_{tx}}{n_p}$ and reliability in different methods.

TABLE II
COMPARISON OF ENERGY CONSUMPTION AND BROADCAST RELIABILITY IN DIFFERENT BROADCAST SCHEMES. THE VALUE OF $p$ IS CHOSEN AS 0.7, 0.37, AND 0.25 FOR PBCAST, RBCAST AND CRBCAST, RESPECTIVELY.

| Broadcasting Scheme | $\frac{N_{tx}}{n_p}$ | Reliability |
|---|---|---|
| Flooding | 10001 | 1 |
| PBcast | 6999 | 0.99 |
| RBcast | 4850 | 1 |
| CRBcast | 2769 | 1 |

We also implement the approximation algorithm for finding an MCDS proposed in [28], which is a centralized algorithm and needs full knowledge about the topology of the network. The algorithm results in 1956 transmissions per packet, which outperforms CRBcast. However, the algorithm in [28] is neither scalable nor practical for large networks. Moreover, it has the uneven load-balancing problem (i.e., some nodes run out of battery power much faster than the others). More importantly, broadcast schemes that are based on finding an MCDS cannot be adapted easily for mobile or lossy networks.

To examine the performance of CRBcast in different network topologies, we further considered two network topolo-

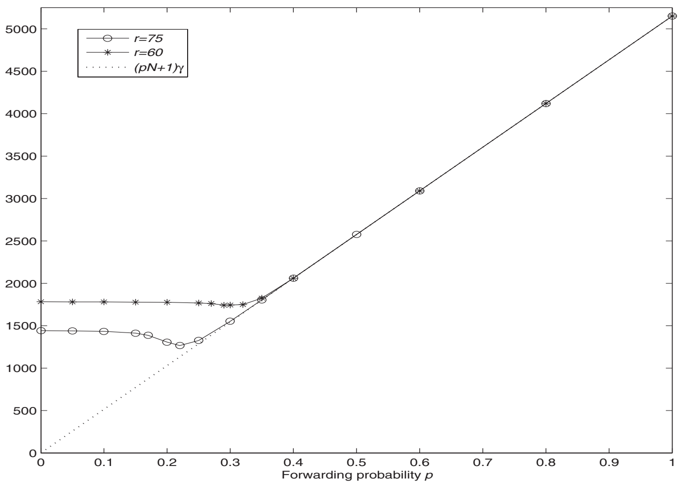


Fig. 6. The number of transmissions per packet for providing reliability at least $1 - 10^{-8}$ in CRBcast. We considered $n_p = 2000$ and $\gamma = 1.03$ for the network topologies $\mathcal{T}_2$ and $\mathcal{T}_3$.

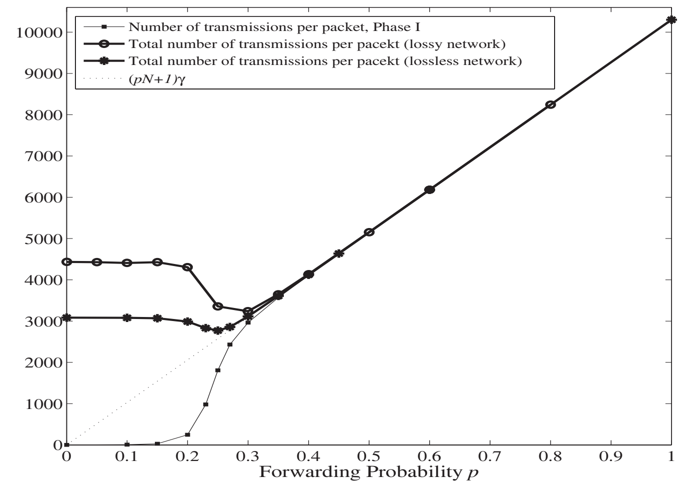


Fig. 7. The number of transmissions per packet for providing reliability at least $1 - 10^{-8}$ in CRBcast. We considered $n_p = 2000$ and $\gamma = 1.03$ for the network topology $\mathcal{T}_1$for lossless links and lossy links with packet loss probability $\epsilon = 0.3$.

gies $\mathcal{T}_2$ and $\mathcal{T}_3$, for which $N = 5000$ nodes are deployed uniformly at random in an area $A = 2000m \times 2000m$. The transmission ranges of the nodes in $\mathcal{T}_2$ and $\mathcal{T}_3$ are $r = 60m$ and $r = 75m$, respectively. Figure 6 depicts the simulation results for broadcasting $n_p = 2000$ packets over $\mathcal{T}_2$ and $\mathcal{T}_3$ using CRBcast. The results are averaged over 50 random network instances for each topology. For each case, we notice an optimal forwarding probability $p^*$ at which the number of transmissions is minimized. From the figure, we have $p_2^* = 0.22$ (corresponding to $\frac{N_{tx}}{n_p} \approx 1267$) and $p_3^* = 0.3$ (corresponding to $\frac{N_{tx}}{n_p} \approx 1742$) for topologies $\mathcal{T}_2$ and $\mathcal{T}_3$, respectively. It should be noted that these results are once again confirmed to be very close to theoretical results for $p^G$ given by (11) ($p_2^G \simeq 0.205$ and $p_3^G \simeq 0.32$). We also compared CRBcast with PBcast. A highly reliable PBcast ($R_{n_p} \geq 0.99$) results in $\frac{N_{tx}}{n_p} \approx 3250$ and $\frac{N_{tx}}{n_p} \approx 4500$ for $\mathcal{T}_2$ and $\mathcal{T}_3$, respectively. Therefore, CRBcast offers about 61% reduction in transmission when compared to PBcast in both $\mathcal{T}_2$ and $\mathcal{T}_3$.

The centralized approximation algorithm for finding an MCDS proposed in [28], results in 903 and 1311 transmissions per packet for $\mathcal{T}_2$ and $\mathcal{T}_3$, respectively. However, as we mentioned earlier such a centralized scheme is not practical.

To evaluate the performance of CRBcast in *lossy* networks, we considered broadcasting $n_p = 2000$ packets over network topology $\mathcal{T}_1$ and link packet loss probability $\epsilon = 0.3$. Figure 7 depicts the results in comparison with the lossless case. As we expect, the number of transmissions per packet in Phase I is the same for both lossless and lossy cases. However, for lossy networks there will be more transmissions in Phase II. Therefore, the total number of transmissions increases. This increase is more considerable for small $p$, for which most of the transmission is done in Phase II. At the optimal value, $\frac{N_{tx}}{n_p} \approx 3237$ for the lossy case, which is about 14% higher than the optimal value for lossless case. A highly reliable PBcast ($R_{n_p} \geq 0.99$) results in $\frac{N_{tx}}{n_p} \approx 7000^{\dagger}$. It should be noted that in less dense networks, PBcast may not even be able to provide reliability close to one when the links are lossy. In contrast, the modified CRBcast for lossy networks will always guarantee reliable delivery of the packets.

### *C. Simulation Results for CRBcast: Time-Constrained Implementation*

In our simulations in Section IV-B, we made some assumptions that may not be necessary or very practical. The first assumption disregarded the waiting time to access the channel. The second assumption in our simulations of CRBcast was that Phase $I$ and Phase $II$ are executed sequentially. In other words, we assumed that Phase $II$ starts when Phase $I$ is over, i.e., all the nodes have already forwarded their packets in Phase $I$. Using these assumptions, we were able to carry out theoretical work and find the optimal forwarding probability. We can argue that these assumptions could be realized if we do not have any constraint on the latency of broadcasting, i.e., if we give broadcasting sufficient time for completion. We refer to this implementation of CRBcast as *time-relaxed* implementation.

In this section, we repeat the experiment assuming the nodes that have some packets to send will *contend* for a channel to avoid any collision. For this purpose, we consider the *two-hop blocking MAC* scheme. We assume that the complete transmission of one packet from a node to its one-hop neighbors takes one time unit. By considering such a MAC layer, we will also be able to compare the latency of the broadcasting algorithms.

Bringing the MAC into the picture, slightly changes the implementation of CRBcast in that we no longer can assume the temporal separation of Phase $I$ and Phase $II$. This means while some nodes may be in the probabilistic forwarding

$^{\dagger}$This result is the same as the result for reliable PBcast in lossless network $\mathcal{T}_1$ (Table II). The reason is that here the network is very dense. Therefore, for high forwarding probabilities, the lost packets are compensated by the redundant packets from different neighbors.

TABLE III
COMPARISON OF ENERGY CONSUMPTION AND LATENCY IN DIFFERENT BROADCAST SCHEMES WITH TWO-HOP BLOCKING MAC IN PLACE. THE VALUE OF $p$ IS CHOSEN AS 0.7 AND 0.27 FOR PBCAST AND CRBCAST, RESPECTIVELY.

| Broadcasting Scheme | $\frac{N_{tx}}{n_p}$ | Latency (time unit) |
|---|---|---|
| Flooding | 10001 | $9.7 \times 10^4$ |
| PBcast | 6999 | $6.7 \times 10^4$ |
| CRBcast | 3053 | $6.1 \times 10^4$ |
| MPR | 8311 | $8.0 \times 10^4$ |

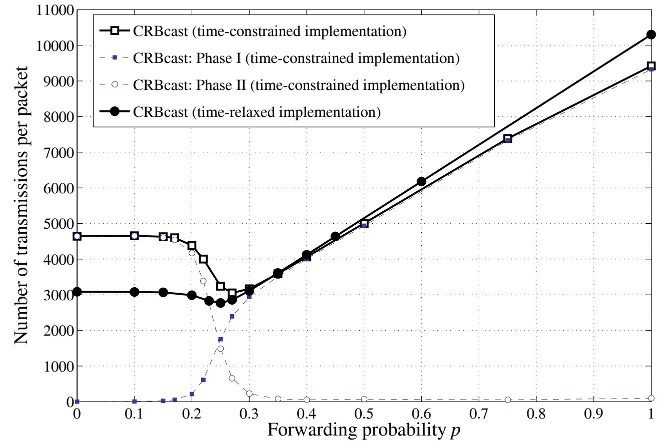


Fig. 8. $\frac{N_{tx}}{n_p}$ versus forwarding probability $p$ for CRBcast over the network topology $\mathcal{T}_1$ for two different implementations.

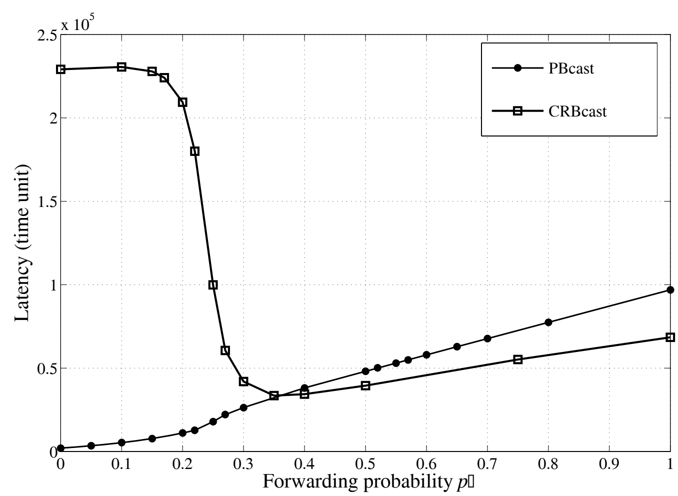


Fig. 9. Latency of PBcast and CRBcast versus forwarding probability $p$ for broadcasting $n_p = 2000$ packets over the network topology $\mathcal{T}_1$. The two-hop blocking MAC was considered for both schemes.

phase, others may have proceeded to Phase $II$. A node is in Phase $I$ at the beginning, and it forwards the packets that it receives for the first time with a probability $p$, whenever it gets the channel. If the channel is not free, it will put the packets in its queue and send one packet at a time, whenever the channel becomes available. A node is considered to be in Phase $II$ if it is either complete or is a neighbor of a complete node (if it hears an ADV message). In these cases, the node will not continue to send those packets that are waiting in its queue. Instead, if it is a complete node, it will send new packets (after decoding and re-encoding) to its neighbors based on the requests that it receives. If it is a neighbor of a complete node, it will request the number of packets it requires to be complete. Upon reception of the required number of packets, it becomes a complete node, and when it has the channel sends an ADV message. This process continues until all the nodes become complete.

We compare CRBcast, PBcast, and another scheme called *Multipoint Relaying* (MPR) [2]. In MPR, each node $i$ is aware of its neighborhood up to two hops and selects a subset $\mathcal{M}_i$ of its one-hop neighbors as forwarding nodes such that for any node $j$ that is two hops away from $i$, there exists a node in $\mathcal{M}_i$ that is connected to $j$. Therefore, if $i$ transmits a packet and only nodes in $\mathcal{M}_i$ forward it, all the nodes in the two-hop neighborhood of $i$ will receive the packet. Yet, when broadcasting is performed, a node $k$ forwards a message received from node $i$ if and only if $k \in \mathcal{M}_i$.

Figure 8 compares $\frac{N_{tx}}{n_p}$ with respect to forwarding probability $p$ when $n_p = 2000$ packets are broadcast over the network topology $\mathcal{T}_1$ using CRBcast for both *time-relaxed* and *time-constrained* cases. We see that there is discrepancy between the two graphs. This is because of different simulation setups. For example, we see that for large values of $p$, time-constrained CRBcast has smaller $\frac{N_{tx}}{n_p}$. This is because there are some nodes in the network that start their Phase $II$ before completing their Phase $I$. As can be seen, similar to the time-relaxed case, there is an optimal forwarding probability for the time-constrained case, and the corresponding optimal forwarding probability values are close. The minimum $\frac{N_{tx}}{n_p}$, which is equal to 3053 transmissions per packet, occurs at $p = 0.27$.

Figure 9 depicts the latency of PBcast and CRBcast versus forwarding probability $p$ for broadcasting $n_p = 2000$ packets over the network topology $\mathcal{T}_1$. The two-hop blocking MAC was considered for both schemes. Clearly, latency is an increasing function of $p$ for PBcast. The case is different for CRBcast. When $p$ is very small, each packet is forwarded by very few nodes in Phase $I$. Therefore, most of the load will be in Phase $II$, which is more time consuming. This causes a high latency. By increasing $p$, a better balance between Phase $I$ and Phase $II$ is in place. Hence, we expect that the latency would decrease since more nodes can forward the packets simultaneously. However, we should note that increasing $p$ beyond a threshold causes many unnecessary transmissions, which increases the latency. This explains the variation of the latency of CRBcast in Figure 9.

Table III summarizes the results for the minimum number of transmissions per packet and the corresponding latency for different broadcast schemes when the two-hop blocking MAC is considered. As can be seen, CRBcast has $69.5\%$, $56.4\%$, and $63.3\%$ less energy consumption in comparison with flooding, PBcast ($p = 0.7$), and MPR, respectively. In terms of latency, PBcast and CRBcast are quite close, though CRBcast has slightly lower latency.

## V. CONCLUSION

In this paper, we considered the problem of reliable and energy-efficient broadcasting in wireless sensor networks. We proposed the collaborative rateless broadcast (CRBcast) protocol, which is based on probabilistic broadcast (PBcast) and rateless erasure coding. In CRBcast, after rateless encoding at the source, a light-weight PBcast is implemented on encoded packets. Then, those nodes that receive sufficient number of packets for retrieving the original data help their neighboring nodes to recover data based on an advertisement and request mechanism. In order to fully understand CRBcast, we investigated PBcast for large networks. We derived an analytical expression for the threshold forwarding probability ($p^{th}$) that ensures reliable broadcasting of a single packet. Moreover, we derived bounds on reliability of PBcast while broadcasting a large number of packets.

Simulation results confirmed that CRBcast substantially reduces energy consumption for reliable broadcasting in comparison with other schemes such as flooding, PBcast, RBcast, and MPR. Specifically, we showed that CRBcast with a reliability of one results in $60\%$ less energy consumption in comparison with PBcast having reliability of $99\%$.

A very important property of CRBcast is that it is not only a reliable and energy-efficient scheme, but also a scalable and practical one that does not require any information about the network topology. Hence, CRBcast can be easily generalized for mobile and lossy networks.

## APPENDIX A
## REVIEW OF RATELESS CODES

Here, we briefly review rateless codes introduced by Luby [17]. Suppose we want to transmit a message comprising of $n$ input symbols. Encoding (output) symbols are simply formed as follows:

- Randomly choose a degree $d$ according to a carefully designed degree distribution
- Choose uniformly at random $d$ input symbols
- Perform bitwise XOR operations on the selected $d$ input symbols to form the output symbol

A receiver is able to retrieve the original data if it receives sufficient number of output symbols. In general, the number of output symbols required to give a high probability of decoding $n$ input symbols can be expressed as $\gamma n$ for a fraction $\gamma \gtrsim 1$ ($\gamma$ is called the rateless overhead). The process of decoding rateless codes relies on finding an output symbol such that the value of all but one of its neighbor input symbols is known. The value of the unknown input symbol is computed by simple bitwise XOR operations. This step is repeated until no more of such output symbols can be found. Following [16] and [29] we may view the input and output symbols as vertices of a bipartite graph $G$. Figure 10 depicts a small example of an LT code, where $n = 7$ and $\gamma = 8/7$. Circular nodes correspond to the input symbols, and the rectangular nodes correspond to the output symbols. The values of the output symbols are known at the receiver, and the goal is to find the values of the input symbols. The decoding starts by copying the value of $c_3$ to its unique neighbor $v_2$. Next, since $c_2$ has only one unknown neighbor, it recovers the value of $v_1$. The next output symbol with only one unknown neighbor is $c_4$ and recovers $v_6$. The decoding continues until no output symbol with exactly one unknown neighbor exists. In this example, the decoding is successful since the values of all the input symbols are determined.

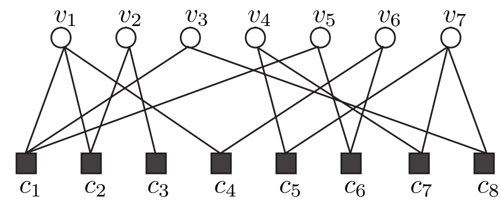


Fig. 10. Example of a small rateless code, where $n = 7$ and $\gamma = 8/7$. The circular and rectangular node correspond to input and output symbols, respectively.

More efficient rateless codes were later proposed in [16], [18]. The ideas are similar to [17], except that a more practical degree distribution is employed to further reduce the encoding and decoding complexities. These codes also use a pre-coder to reduce the decoding failure probability.

## REFERENCES

[1] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: part–II: the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution," *IEEE Trans. Commun.*, vol. 23, no. 12, pp. 1417-1433, 1975.

[2] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast message in mobile wireless networks," in *Proc. 35th Annual Hawaii Int. Conf. System Science*, 2002.

[3] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile Networks Applications*, vol. 7, pp. 481-492, 2002.

[4] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for reliable multicast in multi-hop ad hoc networks," in *Proc. Int. Workshop Discrete Algorithms Methods Mobile Computing Commun.*, pp. 64-71, 1999.

[5] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. 5th Annual ACM/IEEE Int. Conf. Mobile Computing Networking*, pp. 151-162, 1999.

[6] M. R. Garcy and D. S. Johnson, *Computers and Tractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman & Company, 1979.

[7] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proc. IEEE INFOCOM*, pp. 1707-1716, June 2002.

[8] K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky, "Bimodal multicast," *ACM Trans. Computer Syst.*, vol. 17, pp. 41-88, May 1999.

[9] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Trans. Parallel Distributed Syst.*, vol. 13, pp. 14-25, Jan. 2002.

[10] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proc. 3rd ACM International Workshop Modeling, Analysis Simulation Wireless Mobile Systems*, pp. 61-68, Aug. 2000.

[11] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1204-1216, July 2000.

[12] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho, and E. Ahmed, "Minimum-cost multicast over coded packet networks," submitted to *IEEE Trans. Inform. Theory*, 2006.

[13] D. S. Lun, M. Medard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. Allerton Conf. Commun., Control, Computing*, 2004.

[14] D. S. Lun, M. Medard, and R. Koetter, "Efficient operation of wireless packets networks using network coding," in *Proc. International Workshop Convergent Technol.*, 2005.

[15] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," in *Proc. 24th Annual Joint Conf. IEEE Computer Commun. Societies, INFOCOM*, vol. 3, Mar. 2005.

[16] P. Maymounkov, "Online codes," *NYU Technical Report TR2003-883*, 2002.

[17] M. Luby, "LT codes," in *Proc. 43rd Annual IEEE Symposium Foundations Computer Science*, 2002.

[18] A. Shokrollahi, “Raptor codes," *IEEE Trans. Inform. Theory*, vol. 52, pp. 2551-2567, June 2006.

[19] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, “A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM,* Vancouver, BC, Canada, pp. 56-67, Aug. 1998.

[20] J. W. Byers, M. Luby, and M. Mitzenmacher, “A digital fountain approach to asynchronous reliable multicast," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 1528-1540, Oct. 2002.

[21] M. Mitzenmacher, “Digital fountains: A survey and look forward," in *Proc. Inform. Theory Workshop*, pp. 271-276, Oct. 2004.

[22] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. ACM International Conf. Mobile Computing Networking (MOBICOM'99)*, 1999.

[23] Y. Sasson, D. Cavin, and A. Schiper, “Probabilistic broadcast for flooding in wireless mobile ad hoc networks," *Technical Report, Swiss Federal Institute of Technology*, 2002.

[24] B. Krishnamachari, S. B. Wicker, and R. Bejar, “Phase transition phenomena in wireless ad-hoc networks," in *Proc. IEEE GLOBECOM*, San Antonio, TX, Nov. 2001.

[25] N. Rahnavard and F. Fekri, “CRBcast: A collaborative rateless scheme for reliable and energy-efficient broadcasting in wireless sensor networks," in *Proc. 5th ACM/IEEE International Conf. Inform. Processing Sensor Networks*, Nashville, TN, pp. 276-283, Apr. 2006.

[26] P. Gupta and P. R. Kumar, “Critical power for asymptotic connectivity in wireless networks," *Stochastic Analysis, Control, Optimization and Applications: A volume in Honor of W. H. Fleming, W. M. McEneaney, G. Yin and Q. Zhang (Eds.)*, 1998.

[27] H. Pishro-Nik, K. Chan, and F. Fekri, “On connectivity properties of large-scale wireless sensor networks," in *Proc. First Annual IEEE International Conf. Sensor Ad Hoc Commun. Networks*, pp. 498-507, Oct. 2004.

[28] M. V. Marathe, H. Breu, H. B. Hunt-III, S. S. Ravi, and D. J. Rosenkrantz, “Simple heuristics for unit disk graphs," *Networks*, pp. 59-68, 1995.

[29] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, “Practical loss-resilient codes," in *Proc. 29th Annual ACM Symposium Theory Computing (STOC)*, pp. 150-159, 1997.


**Nazanin Rahnavard** (S'97-M'08) received her B.S. and M.S. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 1999 and 2001, respectively. She then joined the Georgia Institute of Technology, Atlanta, GA, in 2002 where she received her Ph.D. degree in the School of Electrical and Computer Engineering in 2007.

Dr. Rahnavard is now with the School of Electrical and Computer Engineering at Oklahoma State University as an Assistant Professor starting from August 2007. Her current research interests lie in the area of telecommunications with a special focus on wireless ad-hoc and sensor networks, cognitive radio networks, and modern error-control coding schemes and their applications. She is the recipient of Outstanding Research Award from the Center for Signal and Image Processing at the Georgia Institute of Technology in 2007

**Badri N. Vellambi** was born in Indore, India, in 1982. He received his Bachelor of Technology degree in electrical engineering from the Indian Institute of Technology, Madras, in 2002 and the M.S. degree in Electrical Engineering and Mathematics from the Georgia Institute of Technology in 2005 and 2008, respectively. He completed his doctoral studies at the Georgia Institute of Technology in 2008. He is currently a research fellow at the Institute for Telecommunications Research, University of South Australia. His research interests include information theory, error control coding, and communication theory.

**Faramarz Fekri** received Ph.D. degree from the Georgia Institute of Technology in 2000. Since 2000, Dr. Fekri has been with the faculty of the School of Electrical and Computer Engineering at the Georgia Institute of Technology where he currently holds an Associate Professor position. He serves on the editorial board of the IEEE TRANSACTIONS ON COMMUNICATIONS, ELSEVIER JOURNAL ON PHYCOM, and on the Technical Program Committees of several IEEE conferences.

Prof. Fekri's current research interests are in the area of communications and signal processing, in particular coding and information theory, information processing for wireless and sensor networks, and communication security. Dr. Fekri received the National Science Foundation CAREER Award (2001), and Southern Center for Electrical Engineering Education (SCEEE) Research Initiation Award (2003), Outstanding Young faculty Award of the School of ECE (2006). He is a Senior Member of the IEEE.