Trade-off between Redundancy and Feedback in Wireless Network Communication

POUYA OSTOVARI[1⋆], ABDALLAH KHREISHAH[2†], JIE WU[1‡], WEI-SHIH YANG[3¶]

[1] *Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122*
[2] *Department of Electrical & Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102*
[3] *Department of Mathematics, Temple University, Philadelphia, PA 19122*

Feedback is an important control mechanism that provides reliability in most wireless network protocols. However, feedback incurs some overhead, especially in lossy network environments. Many previous works on reliable communication neglect the cost of the feedback messages. In this paper, we study the problem of minimum-cost reliable transmission over error-prone wireless networks by considering the cost of feedback. We address two cases: the case where we have a finite number of packets to send and the case where we have infinite packets. In both cases, we provide a solution to the problem with one-hop broadcast transmission. After that, we study the case where network coding is used in our proposed methods. In addition to that, we extend our approaches to address the problem of minimum-cost reliable broadcasting in multi-hop wireless networks. Our simulation results show that the cost of our proposed method is about 40% less than that of the traditional Automatic Repeat reQuest (ARQ) method. Also, the cost of our proposed method with network coding is about 40% less than that of the traditional ARQ

⋆ email: `ostovari@temple.edu`
† email: `abdallah.khreishah@njit.edu`
‡ email: `jiewu@temple.edu`
¶ email: `yang@temple.edu`

method with network coding. We also show that, in the case with small batches of packets, our proposed methods are more efficient than the LT code, which is a rateless code.

Key words: Reliable transmission, feedback, broadcasting, network coding, energy-efficiency, wireless networks.

1 INTRODUCTION

Broadcasting is an important mechanism for disseminating data and control messages in wireless networks. In these applications, all of the sent packets from the source node must be correctly received by every destination node. However, in wireless networks, links are lossy and we need to use certain mechanisms, such as feedback messages, to provide reliability. Automatic Repeat reQuest (ARQ) is the most frequently used approach for addressing packet loss [1]. However, ARQ requires a lot of feedback messages, especially for the case when we have many destination nodes. Hybrid-ARQ methods [2, 3], which combine FEC (Forward Error Correction) with ARQ, are proposed to solve this problem. The RMDP approach [3] uses Vandermonde [4] code and ARQ to ensure reliability. However, the complexity of Vandermonde code is more than that of the XOR coding, which is used in this paper.

Assume that the cost of feedback is negligible. In this case, the source node sends the packets, and then stops to receive feedback from the receivers. Then, in the next iteration, the source node retransmits the lost packets. In this approach, the source node has accurate knowledge about the missing packets before retransmitting them. However, in reality, the cost of feedback messages is not negligible, and in the case of non-zero cost for the feedback, it is possible that sending redundant transmissions before receiving the feedback messages decreases the total cost. The details will be described in the next paragraph.

Assume that we have a source that wants to send 50 packets to 50 destination nodes. Also, assume that the delivery rates of the links between the source and destinations are all 50%. The cost of each sent packet and each feedback is 1, and one feedback message can report all of the missing packets by a destination node. Our goal is to minimize the total number of transmissions, which is equal to the total number of sent packets and feedback messages. Since the links are lossy in this example and there are many destination nodes, we know that to deliver a packet to the destinations, we will

Redundancy level	1	2	3	4	5
Number of sent packets	347	374	398	431	461
Number of feedbacks	298	137	82	57	42
Total transmissions	645	511	480	488	503

TABLE 1
Total number of transmissions based on a simple simulation.

most likely need more than one transmission. Thus, it seems to be logical to send the packets more than once before receiving the feedback. Table 1 shows the average number of transmissions and feedback messages in the cases where the redundancy level varies from 1 to 5. Redundancy level is the number of times we transmit each packet before receiving the feedback. The results in this table are based on the average output of 100 simulation runs. Also, for simplicity, we assume a fixed redundancy level in all of the iterations.

In Table 1, when the redundancy level is equal to 1 the number of sent packets is less than that of the other cases, since the source node does not send redundant packets blindly. However, the number of feedback messages and the total transmissions is much more than in the other cases. The reason is that redundant transmissions increases the probability of receiving the packets by the destination nodes. Therefore, the number of required retransmission iterations decreases, which decreases the number of feedback messages. In this table, the total number of transmissions decreases as we increase the redundancy level from 1 to 3, but after that point, the total number of transmissions starts to increase. Thus, under these settings, it is more efficient to send the missed packet three times before receiving feedback. For highly reliable links, it is likely that we will not need redundancy, but as reliability decreases, redundant transmissions decrease the total cost. Also, the number of feedback messages increases as we increase the number of destination nodes. As a result, more redundancy will be required to decrease the transmission cost. In this example, the redundancy level is fixed for all retransmission iterations. However, changing the redundancy level in different iterations can result to a more efficient solution. In this paper, our goal is to find these redundancy levels.

In addition to redundant transmissions, network coding [5] can be used to increase the efficiency of the ARQ method. Network coding [6–10] is a mechanism in which mathematical operations are used to mix different packets for the purpose of reducing the number of transmitted packets. To improve the transmission efficiency, the work in [11–14] use network coding in the retransmission phase. These methods combine the lost packets at the different receivers to reduce the number of required retransmissions. Assume that in Figure 1, the source node sents two packets, a and b. Destination nodes d_1 and d_2 only received packets a and b, respectively. Therefore, the source node has to retransmit both of the packets. However, the source node can mix the packets to send a single packet $a \oplus b$. Nodes d_1 and d_2 can retrieve their respective lost packets b and a, by performing $a \oplus (a \oplus b)$ and $b \oplus (a \oplus b)$, respectively.

An efficient way to address reliable transmission over error prone channels is to use rateless (fountain) codes [15, 16]. By using rateless codes, the source node can generate and transmit an unlimited number of encoded packets until every destination receives enough packets to retrieve the original packets. In this scheme, the destination nodes need to collect a sufficient number of packets, regardless of which packets have been lost. Assuming that the number of original packets is k, the number of sufficient coded packets is $N = k \times (1 + \epsilon)$ [15]. Here, ϵ is referred to as the overhead. This means that, in order to decode k packets, a destination node needs to receive $N = k \times (1 + \epsilon)$ coded packets. It can be shown that as $k \to \infty$, the overhead goes to zero [17]. Therefore, rateless codes are very efficient for transmitting a large number of packets, but are inefficient for transmitting a small number of packets. As a result, rateless codes are not appropriate for the delay-sensitive applications which need small batches of packets.

There are some previous work, such as the MORE and CCACK methods, that studied the problem of reliable multicasting in lossy wireless networks with a small feedback overhead. This approaches use opportunistic routing to deliver the data to the destination nodes. In opportunistic routing, every node that overhears a packet can be a potential relay node. The MORE method uses random linear network coding [18] to solve the problem of coordinating the nodes in opportunistic routing. The MORE and CCACK methods are well-known and efficient mechanisms for multicating; however, the decoding complexity of the linear coded packet might be a challenge for some networks, such as wireless sensor networks. For this reason, we avoid using linear network coding in this paper.

In this paper, we study the problem of minimum-cost reliable transmission while considering the cost of feedback. Firstly, we find a solution for the case where one-hop transmissions are performed with one destination. We use

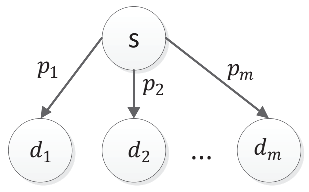


FIGURE 1
One-hop broadcasting.

this result to propose a method for the problem of one-hop broadcasting to multiple destinations. Then, we change the problem to the case with an infinite number of packets, and we propose the optimal solution for the modified problem. In addition, we study the case where network coding is used in our proposed methods. At the end, we extend the proposed one-hop approaches to be used in multi-hop broadcasting applications.

The rest of this paper is organized as follows. In Section 2, we introduce our settings. We propose our methods for one-hop transmission with one destination in Section 3. In Section 4, we use the results from Section 3 to propose our minimum-cost reliable broadcasting methods, and we extend the methods to use network coding. We introduce our multi-hop broadcasting approach in Section 5 and evaluate the proposed methods through simulations in Section 6. Section 7 concludes the paper.

2 SETTING

In this paper, we consider two models. In our first model, the source node wants to broadcast n packets to m destinations (Figure 1). We represent the set of packets as S. The links of the network are lossy, and the delivery rate of the link between the source node and the i-th destination is represented as P_i. The delivery rates can be periodically calculated using probe messages. The source node transmits the batch of packets and receives feedback from each of the destination nodes. The source node uses these feedback messages to find the set of missing packets by the destination nodes, and sends the missing packets in the next iteration. We represent the number of transmissions by the source node, the number of feedback messages, and total cost as t, f, and T, respectively. Our goal is to minimize $T = t + C \times f$, where C is

Notation	Definition
S	The set of packets to be sent
n	Number of packets in S (batch size)
D/m	Set of destination nodes/ Number of destination nodes
d_j	The j-th destination node
g	Number of transmission iterations
f/t	Number of feedback/ Number of packet transmissions
T	Summation of number of transmissions and feedbacks
R	The set of successfully received packets by all destinations
U	Cost of each successful transmission ($U = \frac{T}{R}$)
$P_i, P_{\bar{i}}$	Delivery rate and loss probability of the link between the source node and the i-th destination node
$P_{i\cap\bar{j}}$	Probability of receiving a packet by the i-th destination and not receiving by the j-th destination
k	Redundancy level

TABLE 2
The set of symbols used in this paper.

the cost of each feedback message. We assume that one feedback message can report all of the missing packets by a receiver node, and the cost of each feedback message is equal to the cost of transmitting one packet. Therefore, $T = t + f$, and the objective becomes minimizing the summation of the number of transmissions and feedback messages.

In our second model, the source node broadcasts infinite packets to the destination nodes. In this model, the number of transmitted packets by the source node in each iteration is equal to n. When all of the destination nodes receive a packet, the source node stops the retransmission of that packet. Since the number of packets in this model is very large, our objective becomes minimizing the cost of each successfully received packet by all of the destinations. Our objective is to minimize $U = \frac{f+t}{r}$, where r is the number of successfully received packets. Table 2 summarizes the set of symbols used in this paper.

3 RELIABLE TRANSMISSION TO ONE DESTINATION

Since the cost of the feedback in our model is not zero, in some cases it is more efficient to send redundant packets before receiving the feedback from

Iteration	1	2	3	i
Received packets	0	$n(1-P_{\bar{1}}^k)$	$n(1-P_{\bar{1}}^k)P_{\bar{1}}^k$	$n(1-P_{\bar{1}}^k)(P_{\bar{1}}^k)^{i-1}$
Remaining packets	n	$nP_{\bar{1}}^k$	$n(P_{\bar{1}}^k)^2$	$n(P_{\bar{1}}^k)^i$
Xmissions	kn	$knP_{\bar{1}}^k$	$kn(P_{\bar{1}}^k)^2$	$kn(P_{\bar{1}}^k)^i$

TABLE 3
The number of transmissions, received packets, and remaining packets in different iterations.

the receiver node. In this section, first we keep the transmission redundancy level for all of the retransmission iterations fixed. Then, we discuss why a fixed redundancy level is not optimal, and we modify our method to vary the transmission redundancy level in different retransmission iterations. At the end, we study the second model that represents the case of an infinite number of packets, and we find the optimal solution.

3.1 Finite Packets Case

In our first method, Fix Redundancy for One Destination (FROD), we keep the transmission redundancy level fixed for all of the transmissions. Therefore, in the first iteration, the source node sends n packets k times, where k is the transmission redundancy level, and receives feedback from the destination node. In the next iteration, the source node retransmits the missing packets k times. The source node repeats this process until it finds out that all of the packets have been received by the destination node. To compute the optimal k, we first compute the average number of retransmission iterations, which is obviously equal to the number of feedback messages because, after each iteration, the destination node sends a feedback message. Then, we use this value to compute the number of transmitted packets, which is equal to the summation of the number of transmissions in all iterations. At the end, we find the k that minimizes the summation of the number of feedback messages and transmissions.

The average number of iterations can be calculated as follows:

$$\sum_{i=0}^{f-1} n(1-P_{\bar{1}}^k)(P_{\bar{1}}^k)^i = n \tag{1}$$

Here, f is the number of feedback messages, which is equal to the number

Algorithm 1 Redundancy level for finite packets

```
for a batch of $n$ packets and delivery rate $P_{\bar{1}}$
$k = 1$
while $T'(k) \leq 0$ do
    $k = k + 1$
if $T(k-1) \leq T(k)$ then
    $k = k - 1$
```

of iterations. We represent the delivery and loss probability of the link as P_1 and $P_{\bar{1}}$, respectively. The number of transmissions, received packets, and remaining packets in different iterations are shown in Table 3. Note that both k and n in Equation (1) are integers. In order to find the k that achieves the optimal solution, we relax k and n to be real numbers. Therefore, the right hand side of (1) becomes $n - 0.5$ after relaxation. The left hand side of Equation (1) is a geometric series, so we can rewrite (1) as follows:

$$n(1 - P_{\bar{1}}^k)\frac{1 - P_{\bar{1}}^{kf}}{1 - P_{\bar{1}}^k} = n - 0.5$$

Therefore:

$$f = \log_{P_{\bar{1}}^k}^{\frac{0.5-n}{n}+1} = -\frac{\ln 2n}{k \ln P_{\bar{1}}} \tag{2}$$

Now, we use Equation (2) to compute the number of packet transmissions, which we represent it as t.

$$t = \sum_{i=0}^{f-1} n(P_{\bar{1}}^k)^i$$

Therefore:

$$t = nk\frac{1 - P_{\bar{1}}^{fk}}{1 - P_{\bar{1}}^k} = nk\frac{1 - e^{-\ln(2n)}}{1 - P_{\bar{1}}^k} = k\frac{n - \frac{1}{2}}{1 - P_{\bar{1}}^k}$$

$$T(k) = f + t = -\frac{\ln 2n}{k \ln P_{\bar{1}}} + k\frac{n - \frac{1}{2}}{1 - P_{\bar{1}}^k} \tag{3}$$

Equation (3) has a unique local and absolute minimum (For the proof, refer to the appendix, Proposition 1). Therefore, around the minimizer point, the sign of the first derivative of T changes from negative to positive. To find the optimal k, we start with $k = 1$ and then increase k. For every value of k that we go through, we take the first derivative of T with respect to k. If $\frac{dT}{dk}$ is

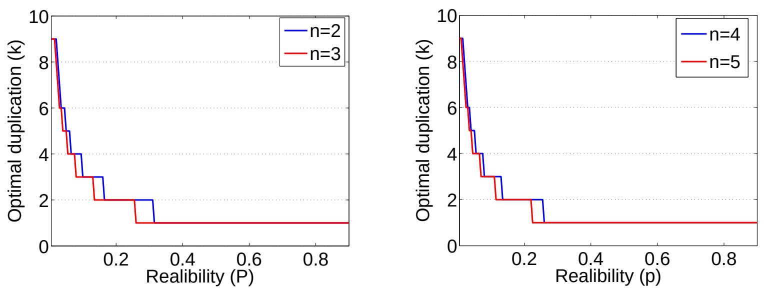


FIGURE 2
Optimal redundancy (k) for $n = 2$ to $n = 5$.

positive, it means that the minimal point is between k and $k - 1$. Therefore, we check the value of $T(k)$ and $T(k-1)$, and we select the minimum value of them. The process of finding the optimal k is shown in Algorithm 1. The optimal k for $n = 2$ to $n = 5$ are shown in Figure 2.

In the FROD method, we use a fixed k for all of the transmission iterations. However, it can be inferred from Figure 2 that for a smaller n, we need more redundancy compared to the greater n. Therefore, for a given n, as more packets are received by the destination node in consecutive iterations, more redundancy for the remaining packets can result in less cost. Thus, in our second proposed approach, we vary the redundancy level, k, in different iterations. We call this approach Changing Redundancy for One Destination (CROD). In CROD, the source node computes the optimal k for n packets based on the FROD approach and sends the packets. Then, the source node receives feedback from the destination; recomputes the new k for the remaining packets and retransmits them. The source node repeats this process until the destination node receives all of the n packets. The CROD method is described in Algorithm 2. In this algorithm, R represents the set of received packets.

3.2 Infinite Packets with Fixed Batch Size

As mentioned in the previous section, since the batch size changes over time, for different iterations, we need different redundancy levels. Therefore, none of the FROD and CROD approaches are optimal. Assuming that the source node has an infinite number of packets to send, we can keep the batch size, n, fixed in different iterations, and we can find the optimal solution for the modified problem. We call this approach Optimal Redundancy for One Des-

Algorithm 2 CROD

```
$R = \{\}$
while $|R| < n$ do
    call Algorithm 1 to find optimal $k$ to transmit $|S - R|$ packets with probability $P_{\bar{1}}$
    transmit the packets in $S - R$, $k$ times
    receive feedback
    update $R$
```

tination (OROD). The source node can be assumed as a router that receives packets at different time slots. In the case where rateless codes are used, the source node needs to wait to receive a large number of packets in order to decrease the overhead of rateless codes. However, this policy increases the delay of the packets. On the other hand, it is not possible to change the batch size during the transmission. Therefore, rateless codes are not appropriate for this scenario. Assume that the number of iterations is g, which goes to infinity. Then, in the case of one destination, the number of feedback messages, f, and the number of packet transmissions will be equal to g and ngk, respectively. Therefore, we have:

$$\begin{aligned} T &= ngk + g \\ r &= ng(1 - P_{\bar{1}}^{k}) \end{aligned}$$

where r is the number of received packets by the destination node in g iterations. Thus, the cost of each received packet will be:

$$U(k) = \frac{ngk + g}{ng(1 - P_{\bar{1}}^{k})} = \frac{nk + 1}{n(1 - P_{\bar{1}}^{k})} \tag{4}$$

which has a unique local and absolute minimum (for the proof, refer to the appendix, Proposition 2). Algorithm 3 shows the approach for finding the optimal k to transmit an infinite number of packets with a fixed batch size equal to n to one destination.

4 RELIABLE BROADCASTING TO m DESTINATIONS

4.1 Finite Packets Case

For m destinations, there are 2^m possibilities for the packet to be received by different destinations. Therefore, after the transmissions, each packet will

Algorithm 3 Redundancy level for infinite packet

```
for a batch of $n$ packets and delivery rate $P_{\bar{1}}$
$k = 1$
while $U'(k) \leq 0$ do
    $k = k + 1$
if $U(k-1) \leq U(k)$ then
    $k = k - 1$
```

belong to one of the 2^m sets that represent the set of nodes that received the packet. For each of these sets, we need to find the optimal redundancy level. On the other hand, the packets that belong to each of the 2^m sets change over time. Because of too many possibilities, it is hard to find the optimal solution. Therefore, we propose the Redundant Broadcasting (RB) heuristic for this problem.

We can use Equation (2) to compute the total number of feedback messages for broadcasting n packets to m destinations. For this purpose, we substitute the average link-loss probability in the equation, and we multiply the result by m. We represent the average link-loss probability as $\bar{P}$.

$$f = -\frac{\ln 2n}{k \ln \bar{P}} m \tag{5}$$

The average number of iterations is equal to $\frac{f}{m} + 1$. The reason is that, when a node receives all of the packets, it will not send any feedback. To estimate the number of packet transmissions in broadcasting n packets to m destinations, we multiply the number of iterations by $\frac{n}{2}$. Therefore:

$$t = (-\frac{\ln 2n}{k \ln \bar{P}} + 1)\frac{n}{2}$$
$$T(k) = -\frac{\ln 2n}{k \ln \bar{P}} m + (1 - \frac{\ln 2n}{k \ln \bar{P}})\frac{n}{2} \tag{6}$$

Equation (6) has a unique local and absolute minimum (The proof is similar to the proof for Equation (3), so for brevity we exclude the proof); we can find its minimal point by checking its derivative. The RB algorithm finds the k that minimizes T and sends the missing packets. In the next iteration, the algorithm recomputes a new k for the remaining packets. The algorithm repeats this process until all of the destinations receive the packets. The RB method is the same as Algorithm 2, but when it calls Algorithm 1, Equation (6) will be used.

Algorithm 4 ERB

```
$R_i = \{\} \forall d_i$
while $|R_i| < n \forall d_i$ do
  for each set $S_j$ do
    $P_j$= average probability of the destinations that have a packet in $S_j$
    In Equation 6 find the optimal $k_j$ to transmit $|S_j|$ packets with probability $P_j$
    transmit the packets in $S_j$, $k_j$ times
  receive feedbacks from the destinations
  update $R_i \forall i$. update $S_j \forall j$
```

The RB algorithm is not efficient since, in each iteration, the redundancy for all of the missed packets is the same. To improve its efficiency, in the Efficient Redundant Broadcasting (ERB) algorithm, we partition the set of packets based on the destinations that missed those packets, and we compute the redundancy level for each partition. After partitioning the set of packets, our algorithm computes the average link-loss probability and the number of packets in each partition. Then, the ERB algorithm uses Equation (6) to compute the redundancy level of each set. Algorithm 4 describes the ERB method. In this algorithm, R_i represents the set of received packets by the i-th destination node. Figure 3 (a) shows the partitions of missing packets by two destinations. In this figure, S_1, S_2, and S_3 are the sets of missing packets by destination d_1, destination d_2, and by both of them, respectively. The binary representation of the sets' indices are shown in the figure. Note that if the i-th bit in the binary representation of an index of a set is 1, destination d_i has a lost packet in that set.

4.2 Infinite Packets with Fixed Batch Size

In the problem of broadcasting an infinite number of packets to m destinations with a fixed batch size, there are 2^m possibilities for the packet to be received by different destination nodes. These states are finite, and the transition probabilities between them only depend on the current states. Therefore, we can use a Markov chain to represent the number of packets in each state and the transitions between the states. Consider a system with 2 destinations, d_1 and d_2, with the link delivery rates of P_1 and P_2, respectively. We use state 11 to represent the packets that have not been received by any destination and state 00 to represent the packets that have been received by both of the destination nodes. We can combine these two states together. Because,

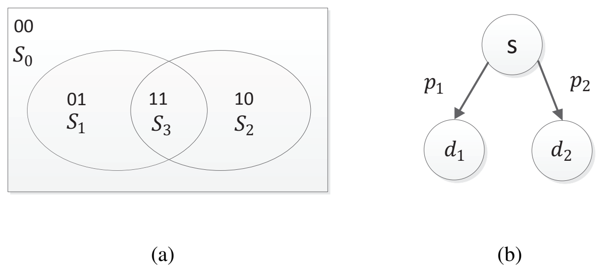


FIGURE 3
(a) Partitions of the missing packets by two destinations. (b) Topology with two destinations.

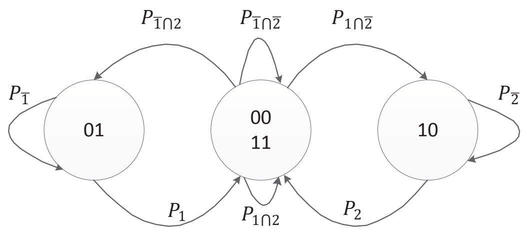


FIGURE 4
Markov chain for two destination nodes.

the batch size is fixed in our model, when a packet is received by all of the destinations, another packet will be added to the batch.

At the beginning, a given packet belongs to state 11, which means it has not been received by any destination. If node d_2 receives this packet and node d_1 does not receive it, the packet will go to state 01. Therefore, the probability of this transition is equal to $P_{\bar{1}\cap 2}$. A packet in state 01 may go to state 00 with probability P_1 and may stay at the current state 01 with probability $P_{\bar{1}}$. Figure 4 shows the Markov chain for the case with two destination nodes.

Assume that the optimal redundancy levels for transmitting the packet in the sets S_1, S_2, and S_3 are k_1, k_2, and k_3, respectively. Here, the indices of the sets show the decimal representation of the states. Because we have a large number of packets, in the steady state condition, the number of incoming packets to each state should be equal to the number of outgoing packets from that state. Therefore, we will have:

$$n_1 P_{\bar{1}} = n_3 P_{\bar{1}\cap 2} \tag{7}$$

$$n_2 P_{\bar{2}} = n_3 P_{1\cap \bar{2}} \tag{8}$$

where n_1, n_2, and n_3 represent the number of packets in states 01, 10, and 11, respectively. Note that the probabilities in these equations are functions of the duplication levels, but for simplicity, we do not show the explicit expressions in the equations. On the other hand, the batch size is fixed and is equal to n. Therefore:

$$n_1 + n_2 + n_3 = n \tag{9}$$

Using Equations 7, 8, and 9 we have:

$$n_1 = \frac{nP_2 P_{\bar{1}\cap 2}}{P_{\bar{1}\cap 2}P_2 + P_{1\cap \bar{2}}P_1 + P_1 P_2}$$
$$n_2 = \frac{nP_1 P_{1\cap \bar{2}}}{P_{\bar{1}\cap 2}P_2 + P_{1\cap \bar{2}}P_1 + P_1 P_2}$$
$$n_3 = \frac{nP_1 P_2}{P_{\bar{1}\cap 2}P_2 + P_{1\cap \bar{2}}P_1 + P_1 P_2}$$

Thus, the number of successfully received packets and the cost of each successfully received packet will be:

$$R = n_1 P_1 + n_2 P_2 + n_3 P_{1\cap 2}$$
$$T = k_1 n_1 + k_2 n_2 + k_3 n_3 + \frac{n}{w}$$

$$U(k) = \frac{T}{R} = \frac{(k_1 + \frac{1}{w})P_{\bar{1}\cap 2}}{P_1 P_{1\cup 2}} + \frac{(k_2 + \frac{1}{w})P_{1\cap \bar{2}}}{P_2 P_{1\cup 2}} + \frac{k_3 + \frac{1}{w}}{P_{1\cup 2}} \tag{10}$$

where, w is the number of packets that can be reported by a single feedback message. We can use mathematical software, such as Matlab, to find the optimal values of k_1, k_2, and k_3.

For more than two destinations, the general approach is the same as when there are two destinations. Firstly, we construct the Markov chain for the problem. Then, we write the steady-state conditions. Finally, we derive the cost function $U(k)$, and we minimize it. The Markov chain with three destination nodes is described in Figure 5.

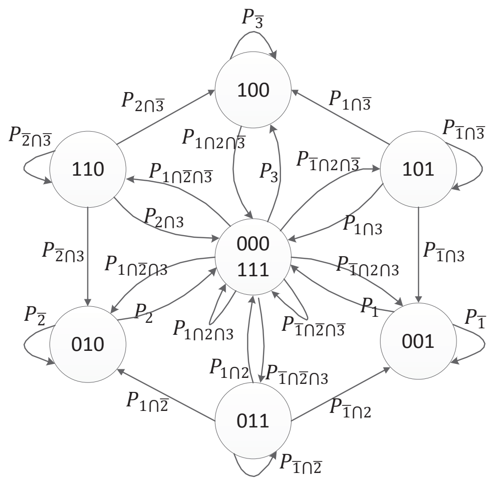


FIGURE 5
Markov chain for three destination nodes.

It can be noticed that finding the optimal solution for the problem of minimum-cost reliable broadcasting with an infinite number of packets is complex since it is exponential in terms of number of destinations. Thus, we propose two heuristics for the problem with multiple destinations. Assume that the number of iterations is g, which goes to infinity. The number of transmissions is equal to $t = ngk$, and the number of feedback messages is equal to $f = mg$. Therefore:

$$T = ngk + mg$$

An estimation for the number of successfully received packets by all of the destinations will be:

$$r = ng(1 - \bar{P}^k)^m$$

where $\bar{P}$ is the average link-loss probability. As a result, the cost of each received packet will be:

$$U(k) = \frac{ngk + mg}{ng(1 - \bar{P}^k)^m} = \frac{nk + m}{n(1 - \bar{P}^k)^m} \tag{11}$$

Equation (11) has a unique local and absolute minimum (the proof is similar to the proof for Equation (4)). Our first heuristic, Redundant Broadcasting

Infinite number of packets (RBI), is similar to the RB method. The difference is that, in the RBI method, we use Equation (11) as the cost function. We have also the Efficient Redundant Broadcasting Infinite number of packets (ERBI) method, which is the same as the ERB approach. However, the ERBI method uses Equation (11) as the cost function.

4.3 Proposed Method With Network Coding

To improve the efficiency of our proposed methods, we can use Network Coding (NC) to decrease the number of transmissions in the retransmission phases. We call these approaches the RB-NC, ERB-NC, RBI-NC, and ERBI-NC approaches. In our approaches, we use instantly decodable network coding [19]. In instantly decodable network coding, a sender node mixes non-coded packets so that its one-hop destinations can decode the coded packet using the received packets in their buffer. This means that the receiver nodes do not need to wait to receive further packets, and they can immediately decode the received coded packets. Therefore, if two packets are missed by the same destination, they cannot be coded together.

The RB-NC method works as follows. Firstly, we use the RB method to compute the redundancy level k. In order to code each packet k times, we set a counter for each missed packet with the initial value equal to k. Then, our algorithm selects one of the lost packets and sequentially checks if there is a packet that can be combined with the selected packets. If there is such a packet, the RB-NC method selects and mix it with the coded packet. Then, the algorihm decreases the counter of the packets by one. The RB-NC repeats this operation to construct other coded packets. When all of the counters become zero, all of the packets are selected k times, so the algorithm stops. The RBI-NC approach is the same as the RB-NC approach, but it uses the RBI approach to compute the redundancy levels. Algorithm 5 describes the ERB-NC approach.

In the ERB and ERBI methods, the redundancy levels of the packets are different. In the ERB-NC and ERBI-NC methods, we calculate the redundancy level of each set S_j by using the ERB and ERBI approaches, respectively. Then, for each set of packets S_j, we assign k_j to all of the packets in that set. Here, k_j is the calculated redundancy level for the packets in set S_j. The rest of the the algorithms are the same as the RB-NC approach.

Algorithm 5 RB-NC

```
set $counter_j = k$ $1 \leq j \leq m$
$i = 1$
**while** exists a $counter > 0$ **do**
    $x_i$ = empty packet
    **for** $j = 1 : m$ **do**
        **if** $counter_j > 0$ **then**
            **if** $x_i \oplus p_j$ is decodable by all destination nodes **then**
                coded packet $x_i = x_i \oplus p_j$
                $counter_j = counter_j - 1$
    $i = i + 1$
```

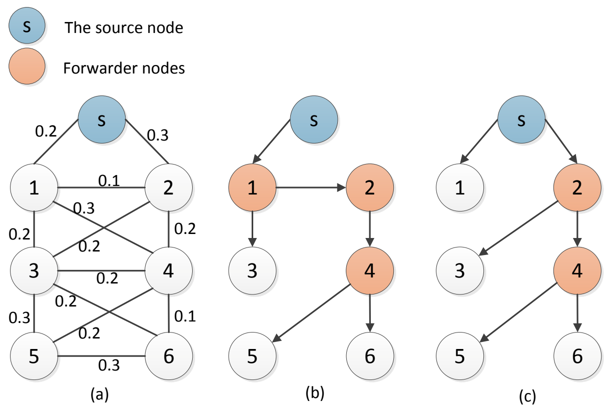


FIGURE 6
(a) A given multi-hop topology. (b) Minimum spanning tree (MST). (c) Fat tree. The loss probability of the links are shown beside the links.

5 MULTI-HOP BROADCASTING

To extend our one-hop broadcasting methods to multi-hop broadcasting, we need to specify the relay nodes. For this purpose, we propose two approaches. Our first approach uses a minimum spanning tree and the second method uses a fat tree to choose the relay nodes.

5.1 Minimum Spanning Tree

In our first approach, we use the link-loss probability as the cost of the links, and we run the distributed version of Prim's algorithm in [20, 21] to build a

Minimum Spanning Tree (MST), rooted at the source node. The result will be a spanning tree with the minimum summation of the loss rates. Then, the source node uses one of the proposed one-hop approaches to broadcast its packets to its children nodes. When a child node that is not a leaf node in the tree receives all of the packets, it uses the same one-hop approach to broadcast the received packets to its child nodes.

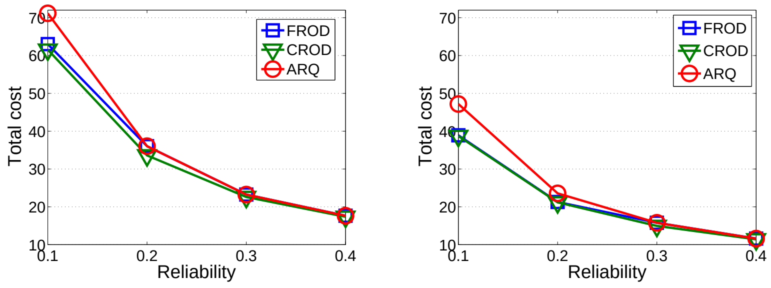
Figure 6 (a) shows a given topology. In this figure, the link-loss probabilities are shown beside the links. The result of the Prim's algorithm is shown in Figure 6 (b). In this figure, node s is the source node, and nodes d_1 and d_3 are the relay nodes. Firstly, node s uses our proposed methods for the one-hop topology to transfer all of its packets to nodes d_1, d_2, and d_3. Then, nodes d_1 and d_3 transfer the received packets to their child nodes.

5.2 Fat Tree

In most topologies, minimum spanning trees have a deep depth. Thus, each relay node will have few number of child nodes, which decreases the efficiency of network coding. To increase the coding efficiency, in our second proposed method for multi-hop transmissions, instead of MST, we use a fat tree to select the relay nodes. A fat tree is a spanning tree with the minimum possible depth. We can construct a fat tree by using the BFS algorithm. The BFS algorithm traverses the tree in a level order fashion. We run the BFT algorithm, and when a node is visited, we connect it to all of its neighbor nodes that are not in the tree. We can use a distributed version of the BFS algorithm, which is introduced in [22]. Figures 6 (b) and (c) show the constructed MST and fat trees of the topology in Figure 6 (a), respectively. It is clear that the depth and number of relay nodes of the constructed MST are more than those of the fat tree. Also, in Figure 6 (c), the source node and node d_2 are the parents of two child nodes, but in Figure 6 (b), they are the parents of one node. Therefore, the coding opportunity in the constructed fat tree is more than that in the MST.

6 SIMULATION RESULTS

In this section, we compare our proposed methods with the traditional ARQ method, in which there is no redundancy, and the source node receives feedback from destinations after sending all of the lost packets. In order to have a fair comparison, we extend the traditional ARQ approach by combining it with network coding. We also compare our proposed method for multiple destinations with LT code [15], which is a rateless code.



(a) $n = 5$ (b) $n = 3$

FIGURE 7
The effect of delivery rate on the total cost during transmission to one destination; $P \in [0.1, 0.4]$.

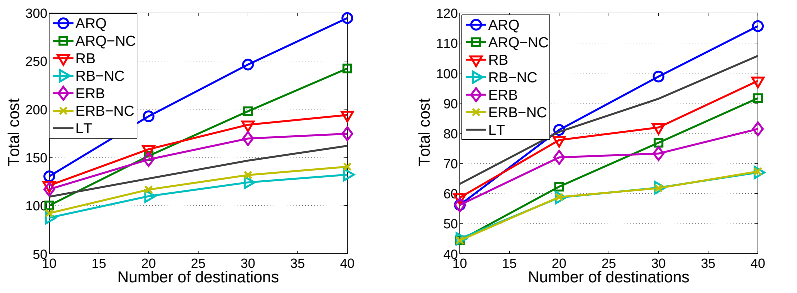
6.1 Simulation Setting

We implemented a simulator in the MATLAB environment to evaluate the proposed methods. We evaluate all of the methods on 1,000 topologies with random link delivery rates. The plots of this paper are based on the average outputs of the simulations. We assume that the delivery rate of the links are independent and the feedback messages are perfect. Also, the cost of each feedback message is equal to one transmission, which means that, for each destination, one feedback message is enough to report all of the lost packets.

In the case of one-hop networks, we assign a random delivery rate to the links between the source and the destination nodes. However, for multi-hop networks, we distribute the nodes in a 10×10 M square area randomly, and we compute the delivery rate of the links based on the Euclidean distance between the nodes. In more details, for any two nodes separated by distance L, we use the Rayleigh fading model [23] to calculate the overhearing probability: $P = \int_{T^*}^{\infty} \frac{2x}{\sigma^2} e^{-\frac{x^2}{\sigma^2}} dx$, where $\sigma^2 \triangleq \frac{1}{(4\pi)^2 L^\alpha}$. We set $\alpha = 2.8$ and the decodable SNR threshold $T^* = 0.02$.

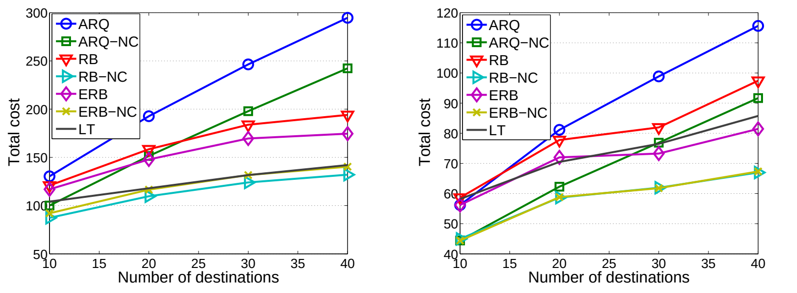
6.2 Simulation Results

In the first experiment, we compare our approaches, CROD (Changing Redundancy for One Destination) and FROD (Fix Redundancy for One Destination), with the ARQ method. In Figures 7 (a) and (b), the delivery rate of the link is in the range of $[0.1, 0.4]$. The batch size n in Figure 7 (a) is equal to 5. It can be inferred from this figure that the CROD method has a smaller total cost compared to that of the FROD method. Also, the total cost



(a) $P \in [0.4, 0.7]$ (b) $P \in [0.7, 1]$

FIGURE 8
Broadcasting to multiple destinations; batch size $n = 20$.

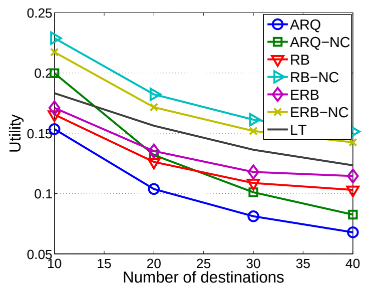


(a) $P \in [0.4, 0.7]$ (b) $P \in [0.7, 1]$

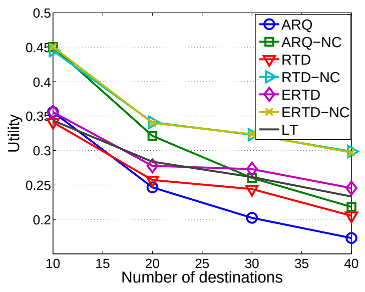
FIGURE 9
Broadcasting to multiple destinations; batch size $n = 20$; cost of ACK is assumed to be half of that of the feedback.

of both of our proposed methods, for one destination, is less than that of the ARQ method. In Figure 7 (b), we decrease the batch size n from 5 to 3. By comparing Figures 7 (a) and (b), we can find that the difference between our approaches and the ARQ method increases as we decrease the batch size.

We compare our approaches for broadcasting a finite number of packets to multiple destinations in Figure 8 (a). In this figure, the delivery rate of the links are in the range of $[0.4, 0.7]$ and $n = 20$. The RB (Redundant Broadcasting) method decreases the cost by up to 33% compared to the ARQ method. Also, the cost of the ERB (Efficient Redundant Broadcasting) approach is



(a) $P \in [0.4, 0.7]$



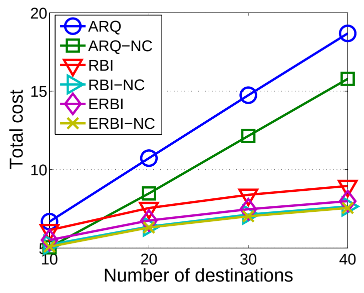
(b) $P \in [0.7, 1]$

FIGURE 10
Broadcasting to multiple destinations; batch size $n = 20$.

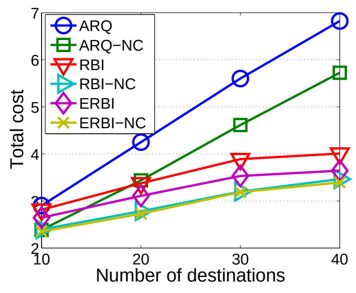
about 40% less than that of the ARQ method. It can be inferred from this figure that the cost of our methods with network coding are about 40% less than that of the ARQ method with network coding. The figure shows that the cost of the LT code method is more than the RB-NC and ERB-NC approaches. The reason is that the overhead of fountain codes for transmitting a small batch of packets is high, so fountain codes are inefficient in this case. It is surprising that in Figure 8 (a), the cost of RB-NC is less than that of the ERB-NC approach. In the RB-NC approach, the redundancy level of all of the packets are the same, which increases the total number of packets to be sent. However, the same redundancy level increases the coding opportunity. Therefore, at the end, the number of transmissions will be close in both approaches, but because of more redundancy, the RB-NC will have a smaller number of retransmission iterations. The reason to increase the delivery rate in this experiment compared to that of the previous experiment is to show that in the case of multi-destinations, even for high delivery rates, our proposed method outperform other approaches.

In Figure 8 (b), we increase the delivery rates to the range of $[0.7, 1]$, and keep $n = 20$. It can be inferred that the cost in Figure 8 (b) is less than that of Figure 8 (a), which is due to the more reliable links. In addition, the LT method becomes less efficient compared to our approaches as the delivery rate of the links increases. This is because of the high overhead of LT codes for transmitting few packets.

In the LT code, the destination nodes send an ACK after receiving enough encoded packets. We assume that the cost of an ACK in the LT method is



(a) $P \in [0.4, 0.7]$



(b) $P \in [0.7, 1]$

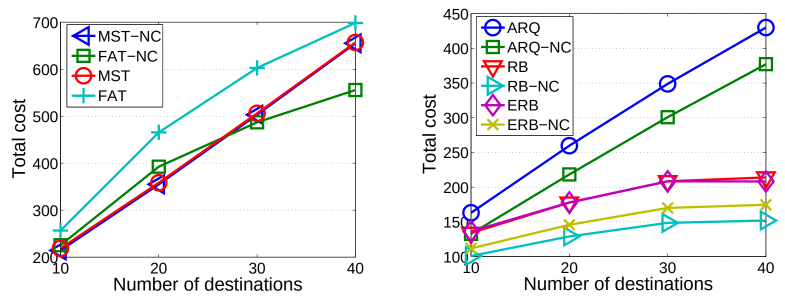
FIGURE 11
Broadcasting infinite number of packets to multiple destinations; batch size $n = 20$.

half of the cost of feedback messages in our approach, and we compare the methods in Figures 9 (a) and (b). The other settings in Figures 9 (a) and (b) are equal to that of Figures 8 (a) and (b), respectively. The simulation results show that the difference between our approaches and the LT codes method is decreased, but the RB-NC and ERB-NC approaches still defeat the LT code approach.

We study the utility of the proposed approaches in Figures 10 (a) and (b). Utility is defined as the division of the number of received original packets by the total transmission time slots. The settings in Figures 10 (a) and (b) are the same as Figures 8 (a) and (b), respectively. The figures show that the utility of the ARQ method is less than the other approaches. Also, the RB-NC and ERB-NC approaches are the most efficient approaches in terms of utility.

We compare our approaches for broadcasting an infinite number of packets to multiple destinations in Figure 11 (a). This figure shows that the RBI (Redundant Broadcasting Infinite number of packets) and ERBI (Efficient Redundant Broadcasting Infinite number of packets) approaches have about 50% less cost compared to that of the ARQ method. Also, the RBI-NC and ERBI-NC approaches are about 55% more efficient than the ARQ method with network coding. In Figure 11 (b), we increase the delivery rates to the range of $[0.7, 1]$. This figure shows that even for highly reliable links, our approaches decrease the cost of transmissions by about 50% compared to the ARQ method and 40% compared to the ARQ method with network coding. As mentioned in Section 3, rateless codes are not appropriate for this case.

Figure 12 shows the cost of the RB and RB-NC method when we apply



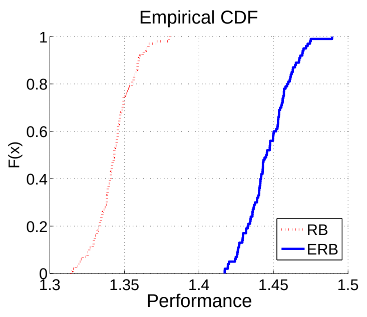
(a) (b)

FIGURE 12
(a) Multi-hop broadcasting; $n = 20$. (b) One-hop broadcasting with lossy feedbacks; $P \in [0.4, 0.7]$; $n = 20$.

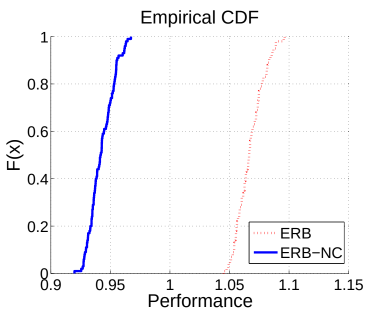
them for multi-hop broadcasting. In this experiment, the batch size n is equal to 20, and the nodes are randomly distributed in a 10×10 M square area. The effect of network coding on the RB method when we use a minimum spanning tree is much less than fat tree. The reason is that the number of neighboring nodes in minimum spanning trees are much less than in fat trees. As stated in the simulation setting section, the network filed size is fixed. Therefore, as we increase the number of nodes, the density of the nodes in the network and the number of relay nodes' neighbors increases. Therefore, the efficiency of network coding increases. That is why for more than 25 nodes, the FAT-NC has less cost compared to the MST-NC method.

We study the effect of feedback loss on the total cost in Figure 12 (b). We assume that the cost of sending an acknowledgment after receiving feedback is negligible. Feedback loss increases the cost of feedback messages. As a result, when the feedback messages are not perfect, the importance of our approaches increases. Figure 12 (b) shows that the RB and ERB approaches decrease the total cost by about 52% compared to the ARQ method. Moreover, the cost of the ERB-NC approach is 60% less than that of the ARQ-NC method.

Figure 13 (a) shows the performance of the RB and ERB approaches for $n = 20$ and $m = 10$. For each simulation run, we calculate the ratio of the total cost in the ARQ and our approaches, and we show the empirical CDF of the results. It can be seen that, in all of the cases, the total cost of our approaches is at least 32% less than the ARQ method. Furthermore, in 40%



(a)



(b)

FIGURE 13
$P \in [0.4, 0.7]$; $n = 20$; $m = 10$. (a) Performance of the RB and ERB approaches over the ARQ method. (b) Performance of the ERB and ERB-NC approaches over the RB and RB-NC method, respectively.

of cases, the performance of the ERB approach is more than 1.45.

In Figure 13 (b), we compare the performance of the ERB and ERB-NC approaches over the RB and RB-NC method, respectively. In this experiment, $n = 20$ and $m = 10$. This figure shows that the total cost in the ERB method is always less than the RB method. In contrast, the ERB-NC method always has a higher cost compared to the RB-NC method.

Figure 14 (a) shows the performance of the FAT and FAT-NC approaches compared to the MST and MST-NC approaches, respectively. In 80% of the cases, the performance of the FAT approach is less than 1. In these cases, the relay nodes have children nodes with high loss rates. In contrast, in the MST approach, the links with the minimum loss rates are selected. This figure shows that network coding decreases the number of cases where the performance is less than one from 80% to 60%. The reason is that, in the FAT approach, each relay node has a higher number of child nodes, which makes network coding more efficient. Therefore, network coding has a larger effect on the FAT approach compared to the MST approach.

Figure 14 (b) shows the effect of delivery rate oscillation on the proposed methods in the case of one-hop transmission. We assign a random delivery rate to the links between the source and the destination nodes, and assume that the delivery rate of each link oscillates around that value. In the figure, we use the center delivery rate of the links as the input of the RTD, RTD-NC, ERTD, and ERTD-NC method. For the RTD*, RTD-NC*, ERTD*, and

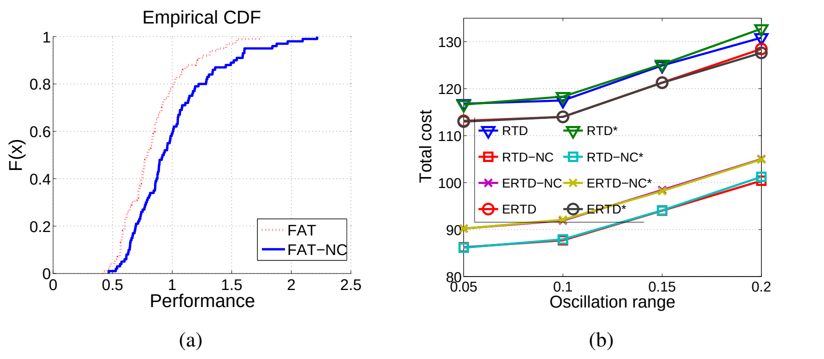


FIGURE 14
(a) Performance od the FAT and FAT-NC approaches over the MST and MST-NC approaches, respectively. $n = 20$; $m = 40$. (b) Effect of delivery rates oscillation on the total cost. $n = 20$; $m = 10$.

ERTD-NC* methods, we consider the oscillation of the delivery rates and use the exact delivery rates. The figure shows that the RTD* method is more efficient than the RTD approach, and as we increase the oscillation range, the difference between the RTD and RTD* increases. Almost the same pattern exists between the other methods. The figure shows that even the oscillation range equal to 0.2 does not have a large effect on the proposed approaches.

7 CONCLUSION

In this paper, we study the problem of minimum-cost reliable broadcasting while considering the cost of feedback messages. We find a solution for one-hop transmission to one destination, and we extend the solution for one-hop broadcasting to multiple destinations. We modify the problem to fit the case of an infinite number of packets, and we show that we can use a Markov chain to find the optimal solution for the modified problem. Then, we study the case where network coding is used in our proposed methods. At the end, we extend the proposed one-hop approach to be used in multi-hop broadcasting applications. Our simulation results show that the cost of our proposed method for the problem of minimum-cost reliable broadcasting is about 40% less than that of the Automatic Repeat reQuest (ARQ) method. Also, the cost of our proposed method with network coding is about 40% less than that of the traditional ARQ method with network coding. Our simulation results show that, in the case with small batches of packets, our proposed methods are more efficient than the LT code.

ACKNOWLEDGMENT

This research was supported in part by NSF grants ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

REFERENCES

[1] H. Djandji, "An efficient hybrid arq protocol for point-to-multipoint communication and its throughput performance," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1688–1698, 1999.

[2] B. Zhao and M. Valenti, "The throughput of hybrid-ARQ protocols for the gaussian collision channel," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1971–1988, 2001.

[3] L. Rizzo and L. Vicisano, "RMDP: an FEC-based reliable multicast protocol for wireless environments," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, no. 2, pp. 23–31, 1998.

[4] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, 1997.

[5] P. Ostovari, J. Wu, and A. Khreishah, *Network Coding Techniques for Wireless and Sensor Networks*. Springer, 2013.

[6] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782– 795, Oct 2003.

[7] S. Biswas and R. Morris, "Exor: opportunistic multi-hop routing for wireless networks," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 133–144, 2005.

[8] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *ACM SIGCOMM*, 2006.

[9] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.

[10] A. Khreishah, I. Khalil, P. Ostovari, and J. Wu, "Flow-based xor network coding for lossy wireless networks," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 2321–2329, 2012.

[11] L. Lu, M. Xiao, M. Skoglund, L. Rasmussen, G. Wu, and S. Li, "Efficient network coding for wireless broadcasting," in *Wireless Communications and Networking Conference (WCNC)*, 2010, pp. 1–6.

[12] L. Lu, M. Xiao, and L. Rasmussen, "Relay-aided broadcasting with instantaneously decodable binary network codes," in *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, 2011, pp. 1–5.

[13] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 914–925, 2009.

[14] W. Fang, F. Liu, Z. Liu, L. Shu, and S. Nishio, "Reliable broadcast transmission in wireless networks based on network coding," in *Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 555–559.

[15] M. Luby, "LT codes," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280.

[16] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[17] P. Cataldi, M. Shatarski, M. Grangetto, and E. Magli, "Lt codes," in *IIH-MSP'06*, 2006, pp. 263–266.

[18] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[19] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive network coding for broadcast channels," in *Workshop on Network Coding, Theory, and Applications (NetCod'09)*, 2009, pp. 80–85.

[20] R. Gallager, P. Humblet, and P. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and systems (TOPLAS)*, vol. 5, no. 1, pp. 66–77, 1983.

[21] Y. Dalal, "A distributed algorithm for constructing minimal spanning trees," *IEEE Transactions on Software Engineering*, no. 3, pp. 398–405, 1987.

[22] B. Awerbuch and R. Gallager, "Distributed BFS algorithms," in *26th Annual Symposium on Foundations of Computer Science*, 1985, pp. 250–256.

[23] C. Wang, A. Khreishah, and N. Shroff, "Cross-layer optimizations for intersession network coding on practical 2-hop relay networks," in *Asilomar*, 2009, pp. 771–775.

Proposition 1. *For $0 < P_{\bar{1}} < 1$, $T(k)$ in (3) has a unique local and absolute minimum at $k_0 \in (0, \infty)$.*

Proof. From (3):

$$T(k) = -\frac{\ln 2n}{k \ln P_{\bar{1}}} + k\frac{n - \frac{1}{2}}{1 - P_{\bar{1}}^k}$$

$-\frac{\ln 2n}{k \ln P_{\bar{1}}}$ is obviously a convex function. Thus, it is sufficient to show that the second part is also a convex function.

$$t = k\frac{n - \frac{1}{2}}{1 - P_{\bar{1}}^k}$$

$$t'(k) = \frac{n - \frac{1}{2}}{(1 - P_{\bar{1}}^k)^2}[(1 - P_{\bar{1}}{}^k) + P_{\bar{1}}^k k \ln P_{\bar{1}}]$$

we write $t'(k)$ as:

$$t'(k) = \frac{(n - \frac{1}{2})P_{\bar{1}}^k}{(1 - P_{\bar{1}}^k)^2} y(a, k)$$

where, $y(a, k) = a^k - [k \ln a + 1]$, such that $a = \frac{1}{P_{\bar{1}}}$

$$\frac{\partial y}{\partial k} = a^k \ln a - \ln a = \ln a(a^k - 1) > 0, \forall k$$

on the other hand:

$$y(a,0) = 0$$
$$\lim_{k\to+\infty} y(a,k) = +\infty$$

$y(a,k)$ is continuous on $(0,+\infty)$, so only at $k_0 = 0$ it is equal to zero. Therefore, function t is also a convex function. □

Proposition 2. *For* $0 < P_{\bar{1}} < 1$, $U(k)$ *(4) has a unique local and absolute minimum at* $k_0 \in (0,\infty)$.

Proof. From (4):

$$U(k) = \frac{nk+1}{n(1-P_{\bar{1}}^k)} = \frac{k}{1-P_{\bar{1}}^k} + \frac{1}{n(1-P_{\bar{1}}^k)}$$

therefore:

$$U'(k) = \frac{1}{(1-P_{\bar{1}}^k)^2}[P_{\bar{1}}^k(\ln P_{\bar{1}}^k)(k+\frac{1}{n}) + (1-P_{\bar{1}}^k)]$$

we write U' as:

$$U'(k) = \frac{P_{\bar{1}}^k}{(1-P_{\bar{1}}^k)^2} y(a,k)$$

where, $y(a,k) = a^k - [k\ln a + \frac{1}{n}\ln a + 1]$, $a = \frac{1}{P_{\bar{1}}}$

$$\frac{\partial y}{\partial k} = a^k \ln a - \ln a = \ln a(a^k - 1) > 0, \forall k$$

on the other hand:

$$y(a,0) = 1 - \frac{1}{n}\ln a - 1 = -\frac{1}{n}\ln a < 0$$
$$\lim_{k\to+\infty} y(a,k) = +\infty$$

$y(a,k)$ is continuous on $(0,+\infty)$, so there is a unique k_0 such that $y(a,k_0) = 0$. □