MULTICASTING IN WIRELESS NETWORKS USING RATELESS CODES AND OPPORTUNISTIC ROUTING

A thesis submitted in partial fulfillment of
the requirements for the degree of

Bachelor of Technology

by

Gagandeep Singh and **G Barath Shankar**

(**11010273** and **11010277**)

Under the guidance of

Prof. Sanjay K. Bose

DEPARTMENT OF ELECTRONICS & ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
April 2015

Certificate

This is to certify that the work contained in this thesis entitled

Multicasting in Wireless Networks using Rateless Codes and Opportunistic Routing

is the work of

Gagandeep Singh and **G Barath Shankar**
(11010273 and 11010277)

for the award of the degree of Bachelor of Technology, carried out in the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.

____

Prof. Sanjay K. Bose

Date:_____________
Place:_____________

ACKNOWLEDGEMENT

We would like to convey our gratitude to Prof. Sanjay K. Bose, who has been the guiding force behind this work, for his invaluable advice and encouragement. We deeply acknowledge his consistent support and motivation in every aspect of academic life. We also take this opportunity to thank Mr. Sonu K. Mishra for his valuable inputs and advice. We thank the entire fraternity of IIT Guwahati for the enormous impact they have had on our lives and the way they have shaped our thoughts. Finally, we thank our parents for their enduring love and support.

ABSTRACT

Rateless Codes, also known as Fountain Codes, are a class of erasure codes with the property that from a given set of source symbols, a potentially unlimited sequence of encoding symbols can be generated such that the original source symbols can be recovered from any subset of the encoding symbols of size equal to or only slightly larger than the number of source symbols. In erasure channels, reliable multicasting requires a lot of feedback. Rateless codes are particularly well suited for reliable multicasting over wireless networks as they efficiently reduce the feedback required. Since multiple neighboring nodes to a transmitter can hear the transmission in a wireless network, opportunistic routing can also be applied to reduce both the number of transmissions needed as well as the transmission time. We use a simple rateless encoding scheme to propose several opportunistic multicast routing algorithms which can be used in a wireless network. The performance of the proposed algorithms is studied using simulations.

Contents

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

N	Number of packets per message
N^*	Mean total number of symbol transmissions required for message delivery across link
M	Number of symbols missing at the receiver
K	Number of independent symbols received at destination
p	Link erasure probability
C_{link}	Link cost
J	Number of potential forwarders considered
L	Number of multicast destinations considered
MNT	Mean number of transmissions per packet
MTT	Mean transmission time for message delivery

Chapter 1

INTRODUCTION

1.1 Motivation

Over the last couple of decades, the number of wireless users has seen a tremendous growth worldwide. This increasing density of wireless networks coupled with requirements for faster data speeds has made it essential to improve the existing routing strategies so as to transmit messages with minimum latency as well as number of transmissions.

Erasure channels are the ones in which a packet is either correctly received or it gets fully lost. When the packet arrives without any error or when the error-correcting code successfully corrects the errors, the packet is assumed to have successfully arrived. On the other hand, when the decoding of the packet fails, the packet with errors is simply assumed to be erased and hence is discarded. In general, reliable transmission over erasure channels involves identifying the erased packets and retransmitting them. However there are some drawbacks in using the usual retransmission schemes. First, the added complexity of introducing a feedback channel does not add to the channel capacity [1]. Secondly, these would be particularly difficult to implement in multicast scenarios as different receivers may have different missing packets resulting in a feedback implosion. This involves substantial feedback and retransmission costs [2].

Rateless codes are erasure codes that offer an elegant alternative using minimal feedback which can nevertheless use the channel very efficiently [3-5]. In rateless approach, a message is divided into several packets and each packet is treated as a symbol and a message consists of a sequence of such symbols, called the source symbols. The basic idea behind rateless codes is that from a given set of source symbols, a potentially unlimited sequence of encoding symbols can be generated such that the original source symbols can be recovered from any subset of the encoding symbols of size equal to or only slightly larger than the number of source symbols. Furthermore encoding symbols can be generated on the fly, as few or as many as required. In this work, we use a simple rateless code approach where a transmitter would transmit symbols which are random linear combinations of the source symbols over the erasure channel until enough combinations have been received to allow the receiver to decode each original symbol and hence recover the message. In such a transmission, when the transmitter sends random linear combinations of packets, it is unaware of which packets have been received and which have been erased. The transmitter node receives an acknowledgement only when the entire message has reached the receiver. Thus a single feedback message is only sent even though several packets were sent. Thus there is a substantial reduction in the number of feedback messages.

In wireless networks, it is often less costly to transmit to at least one node in a set of neighbors than to a specific neighbor. This observation is exploited by opportunistic routing protocols [6][7]. Traditional routing is based on the discovery of a path previously to the transmission of the packet. In opportunistic routing, instead of selecting a node to act as the next hop a-priori, the relay node is determined when the message is being transmitted. In our algorithms, instead of following a predefined least cost path for transmission, opportunistic approach for selecting the next forwarders is used. The advantages of using this approach are that it allows a transmitter to simultaneously transmit over several erasure links and choosing the next forwarders dynamically based on the neighbors that have actually received the message. This gives rise to better progress per transmission towards the destination. And as the channel is wireless, multiple neighboring nodes will anyhow receive the transmission and hence there is no more usage of network capacity in this scenario compared to traditional routing. These protocols generally perform better than fixed path routing in which a single fixed and best path is selected and used for subsequent transmissions.

This work considers the problem of multicast routing when each node of the network uses rateless coding for transmission. Although this paper assumes that the network graph and the erasure rates of each link are fixed, slow changes in the network can be handled by suitably updating this information in the network nodes.

1.2 Literature Survey

There exist a large number of rateless (fountain) coding schemes in literature. We discuss a few of them in this section. The simplest ones are the random linear fountain codes, in which the packets to be transmitted are the random linear combinations of the source message packets. It is assumed that the receiver knows the packets that are present in the given linear combination. If the receiver has received lesser than N packets, where N is the number of packets which constitute the message, then decoding is not possible. If it has received N linearly independent packet combinations, then the receiver can decode the entire message. The problem with this scheme is the high encoding and decoding costs, which are respectively quadratic and cubic in the number of encoded packets [5].

The Luby-Transform (LT) [3] code reduces the encoding and decoding complexities of the random linear fountain codes to $N \log N$. The packets to be transmitted are encoded as follows: A degree d is chosen randomly from a predefined degree distribution which depends on the message size. Then d distinct input packets are chosen uniformly from the source packets, and are EX-ORed bitwise to produce the packet to be transmitted. The degree is chosen such that the resulting generator matrix is sparse, which simplifies the encoding and decoding algorithm. The decoding algorithm is based on simple message passing and is explained in [4]. Occasionally, this decoding algorithm gets stuck due to absence of single link packets, i.e. the packets which are same as the source message packets. We have used the abstraction of the rateless coding and decoding procedures as described in [8][12]. We also borrow the definition of link cost from [8] that is appropriate when message transmission is done using rateless coding.

We now look at some opportunistic routing protocols discussed in the literature. In ExOR (Extremely Opportunistic Routing) protocol is described in [7]. ExOR determines the path as the packet moves through the network, based on which nodes receive each transmission. It proposes a distributed MAC protocol that allows recipients to ensure that only one of them forwards the packet, and an algorithm that predicts which recipient is likely to be the most useful forwarder. [9] and [10] address the Least Cost Anypath Routing (LCAR) problem which is about how to assign a set of candidate relays at each node for a given destination such that the expected cost of forwarding a packet to the destination is minimized. To solve the problem, a generalization of single-path routing, which is called anypath routing is introduced. In anypath routing, the next hop to reach a destination is explicitly treated as a set of neighbors rather than a single neighbor. The notion of single-path route is generalized to that of anypath route, which is the union of all possible packet trajectories induced by an assignment of candidate relays. Under this framework, new any-path cost metrics are proposed to solve the problem instead of using the traditional single-path cost metrics. [11] extends the previously mentioned work to multicast scenarios. It proposes multicast cost metrics that could be calculated in a Bellman-Ford type approach. It uses them to determine the forwarding set and the forwarding strategy so that packets can reach all destination nodes efficiently. [13] proposes a new multicast routing protocol for ad hoc wireless networks called AMRIS. The protocol uses dynamically assigned id-numbers to form a multicast delivery tree of all nodes participating in a multicast session. These id-numbers help the nodes dynamically leave and join a multicast session, as well as adapt rapidly to changes in link connectivity due to mobility.

Chapter 2

RATELESS CODING SCHEME

2.1 Coding Scheme

The rateless coding scheme used by us is based on the approach proposed in [8]. A message to be transmitted is divided into N symbols, where each symbol represents a packet. The rateless code is used to send these N packets over an erasure channel. The receiver retains only the error free transmissions. The process is abstracted as a source transmitting binary vectors of length N, with 1 in the i^{th} position if symbol i is a part of the corresponding combination of symbols being transmitted. The receiver is able to decode the message once it receives N independent binary vectors, and indicates this by sending an acknowledgement packet ACK to the source. The first N transmissions are the vectors (1,0,…,0),(0,1,…,0),(0,0,…,1). The subsequent ones are the linear combinations of these obtained by randomly choosing 1 or 0 for each element of the vector. The all-0 choice is excluded by the transmitter by choosing another vector.

2.2 Link Cost Definition

We consider a scenario in which the receiver has successfully received K independent, non-zero, N-bit binary vectors, where $0 \le K \le N$. If it now receives an N-bit non-zero binary vector with 1/0 equally likely for each bit, then this vector is retained by the receiver if it is independent of the K vectors that it already has. The probability P(N, K) of this is

$$\mathrm{P}(N,K) = \frac{(2^N - 1) - (2^K - 1)}{(2^N - 1)}$$

$$= 1 - \frac{2^K - 1}{2^N - 1} \tag{2.1}$$

Now as expected,

$$\mathrm{P}(N, 0) = 1 \tag{2.2}$$

$$\mathrm{P}(N, 1) = 1 - \frac{1}{2^N - 1} \tag{2.3}$$

$$\mathrm{P}(N, N) = 0 \tag{2.4}$$

The mean number of additional symbol transmissions required after the first N transmissions, given that M symbols out of these N were erased, is calculated by the following recursion

$$\mathrm{E}[M, N, p] = \frac{1}{(1-p)\mathrm{P}(N, N-M)} + \mathrm{E}[M-1, N, p], \tag{2.5}$$

where p is the erasure probability and $\mathrm{E}(0, N, p) = 0$. Therefore, mean number of total symbol transmissions N^* required is given by

$$N^* = N + \sum_{M=1}^{N} \mathrm{E}[M, N, p] \binom{N}{M} p^M (1-p)^{(N-M)} \tag{2.6}$$

We define the corresponding link cost C_{link} as

$$C_{link} = \frac{N^*}{N} \tag{2.7}$$

Chapter 3

ROUTING ALGORITHMS

3.1 Network Model

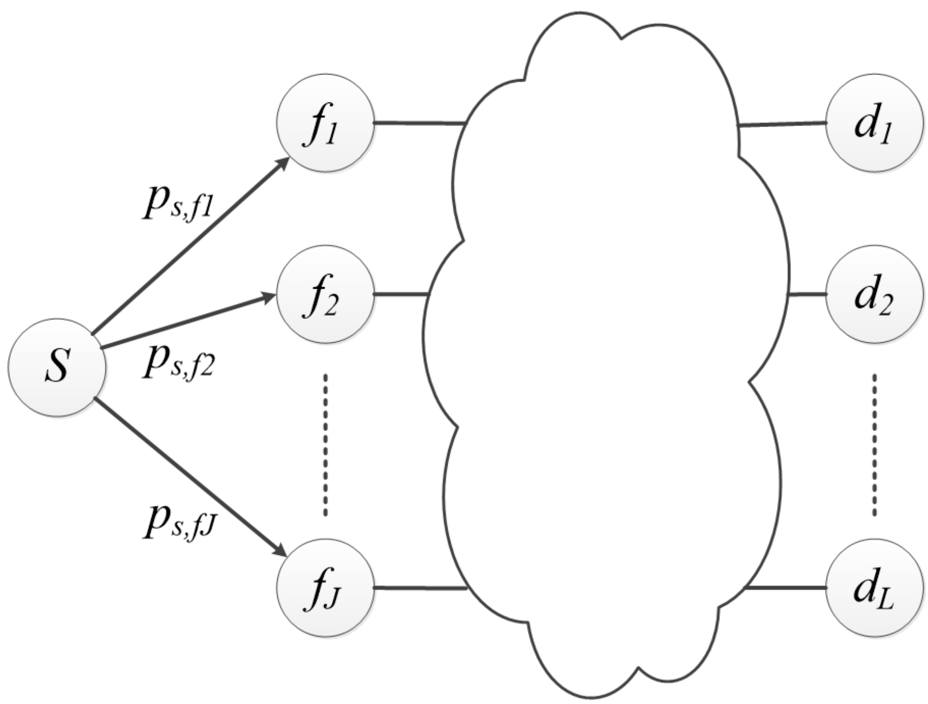

Fig. 3.1 A representative network

We consider the representative network shown in Fig. 1 with source S, destination set D consisting of nodes $\{d_1,d_2,\ldots,d_L\}$ and a set $F = \{f_1,f_2,\ldots,f_J\}$ of neighboring nodes of S with their wireless links S to having symbol erasure probabilities p_{S,f_j} respectively. Two nodes are considered neighbors if the link between them has erasure probability below a certain threshold. Using (2.7) we can calculate the link cost of each of these J links. The cost from forwarder node f_j to a destination node d_l is denoted by C_{f_j,d_l} and is defined as the cost along the minimum unicast cost path from f_j to d_l computed using the standard *Djikstra* or *Bellman Ford* algorithms. In multicasting scenarios, it is possible to define the least cost from any node to each possible subset of the destination set as was done in [11]. We have intentionally avoided this approach because of the high computational overhead and processing that this would require and instead opted to work only with the single least cost paths to each node that can be reached in the destination set for simplicity even though this would lead to marginally poorer performance. The least cost path may be direct or may go through other nodes of the network. Moreover, the least cost path to a destination node may possibly also go through some other node in the destination set. We assume that global link state information is available at each node l. This can be done using a standard *Distributed Bellman-Ford* type approach. For convenience, we consider a static scenario where the network graph and the erasure probability of each link are fixed. However, slow changes in the network can still be handled by suitably updating this information in the network nodes.

3.2 Routing Algorithms

We now present routing algorithms for networks where nodes use rateless codes for transmissions. It is assumed that ACKs, being small in size, arrive without any error; hence the erasures of ACKs have not been considered in designing the protocols. The problems arising from such erasures can anyway be circumvented by taking routing decisions based on the ACKs which have arrived.

3.2.1 Greedy Forwarder Select-Routing Algorithm (*GFS-RA*)

This algorithm is based on a simple greedy approach for selecting the forwarders. At each hop, the transmitting node will maintain a list of all the neighbors which have sent an ACK and then selects the best one in a simple greedy fashion (described subsequently) as the forwarder for each reachable destination in the destination set. The exact working of this algorithm is described next where a source will transmits packets using the rateless scheme as described earlier.

Without loss of generality, let f_1 be the first node to receive the complete message. It then sends an ACK back to the source S. Upon receiving the acknowledgement, S marks all the destination nodes which f_1 can reach with a cost lower than that from S and node f_1 is added to set of acknowledged neighbors. If all the destination nodes have been marked then it will stop the transmission; else it will continue transmitting. Let f_2 be the next node to receive the complete message and send an ACK to S. S then marks all the destination nodes reachable by f_2 and adds f_2 to the set of acknowledged neighbors regardless of whether it can reach any previously unmarked destination or not. It checks again if all the destinations have been marked. If so, it stops transmission; else it continues transmitting and the whole process repeats until all destination nodes have been marked. Once all the nodes in the destination set have been marked in this fashion, then for each destination the source selects that node, from the set of acknowledged neighbors, which has the lowest cost of reaching that destination as the forwarder. At the end of this process we will have a set of destinations for each forwarder. Now source directs each forwarder to begin forwarding to its corresponding destination set. The transmission from any forwarder node to its own destination set will be same as the process just described with that forwarder acting as the source. This is a simple algorithm that we use as a benchmark for the more efficient (but more complex) subsequent algorithms proposed subsequently. Algorithm 1. presents the pseudo code for choosing the forwarders at a transmitting node *S*.

Algorithm 1: Pseudo Code for *GFS-RA*

Input: $D = \{d_1, d_2, \ldots, d_L\}$, $F = \{f_1, f_2, \ldots, f_J\}$

$C_{f_j,d_l}, C_{S,d_l} \forall j,l$

1. Initialize: $D_{sub_j} = \phi \; \forall \; j = 1, 2, \ldots J; U = D; A_{fwd} = \phi$
2. Start message transmission using rateless approach.
3. until (ACK from any f_j)

   $\forall \, d_l : C_{f_j,d_l} < C_{S,d_l}$

   $U = U - d_l$

   $A_{fwd} = A_{fwd} \cup f_j$
4. If ($U == \phi$)

   Stop transmission

   Else

   Continue Transmission and go to 3.
5. $\forall \, d_l$

   $k = \operatorname*{argmin}_{j : f_j \in A_{fwd}} C_{f_j,d_l}$

   $D_{sub_k} = D_{sub_k} \cup d_l$
6. $\forall \, f_j \in A_{fwd} : D_{sub_j} \neq \phi$

   Ask f_j to transmit to D_{sub_j}

3.2.2 Constrained Forwarder Set-Routing Algorithm (*CFS-RA*)

The main goal of *CFS-RA* is to minimize the overall transmission time to deliver a message from the source to all the nodes in the destination set. This is achieved by passing on the message to the next forwarders, at each hop, as quickly as possible without deviating too much from the shortest path. At each hop, we try to restrict the potential forwarder set for each destination by including only those neighbors which can reach the destination with a cost below a certain threshold. The threshold for a destination is set to be equal to α times more than the minimum cost of reaching the destination from amongst all the neighbors. Here the value α is a parameter of the algorithm to be optimally chosen. The algorithm is implemented as follows.

The source S starts transmitting the packets using the rateless scheme as described earlier. Let the node f_1 be the first node to receive the complete message. It then sends an ACK back to the source. Upon receiving the acknowledgement, the source looks at the threshold costs corresponding to all of its destinations and sends a special packet RACK to f_1 requesting it to forward to each such destination where the unicast cost from f_1 to that destination is lower than the threshold cost for that destination as well as the cost from source to that destination. If f_1 does not satisfy the above criteria for any destination, then a negative acknowledgement packet NACK is sent instead. Upon receiving a RACK, f_1 immediately starts its transmission. S then updates the destination set by making a list of all the destination nodes which have been assigned f_1 as the forwarder and removing them from the destination set. If the destination set becomes null then it will stop the transmission else it will continue transmitting. Let f_2 be the next node to receive the complete message. It will send an ACK to S. S replies appropriately either with a NACK, or a RACK directing it to forward the message to the corresponding destinations which belong to the updated destination set, i.e. the set of destination nodes which have not been removed yet. The source will again update the destination set. In case it becomes empty, it stops transmission else it continues transmitting and the whole process repeats. The transmission from any forwarder node to its own destination set will be same as the process just described with that forwarder acting as the source. Algorithm 2 presents the pseudo code for choosing the forwarders at a transmitting node S.

Algorithm 2: Pseudo Code for *CFS-RA*

Input: $D = \{d_1, d_2, \ldots, d_L\}$, $F = \{f_1, f_2, \ldots, f_J\}$

$C_{f_j,d_l}, C_{S,d_l} \forall j, l; \alpha$

1. $C_{th_l} = (1+\alpha) \times \min_j C_{f_j,d_l}$
2. Initialize: $D_{sub_j} = \phi \,\forall\, j = 1, 2, \ldots J; U = D$
3. Start message transmission using rateless approach
4. until (ACK from any f_j)

   $\forall\, d_l : C_{f_j,d_l} < C_{S,d_l} \;\&\; C_{f_j,d_l} \le C_{th_l}$

   $U = U - d_l$

   $D_{sub_j} = D_{sub_j} \cup d_l$
5. If ($D_{sub_j} \ne \phi$)

   Ask f_j to transmit to D_{sub_j}
6. If ($U == \phi$)

   Stop transmission

   Else

   Continue Transmission and go to 4.

3.2.3 Minimum Forwarder Set-Routing Algorithm (*MFS-RA*)

In this algorithm, the main objective is to minimize the number of transmissions. We do this by greedily minimizing the branching i.e. minimize the number of forwarders selected at each hop. This algorithm is similar to *GFS-RA* except that the optimization is done with respect to the number of forwarders chosen at each hop to reach all the destinations. As in the case of *CFS-RA*, we restrict the potential forwarder set for each destination by including only those neighbors which can reach the destination with a cost below a certain threshold. The threshold value is chosen the same way as in *CFS-RA* with a threshold parameter α. The algorithm is implemented as follows.

The source starts transmission using the rateless scheme as described earlier. Let f_1 be the first node to receive the complete message. It then sends an ACK back to the source *S*. Upon receiving the acknowledgement, the source checks the threshold costs corresponding to all of its destinations and marks each such destination node which f_1 can reach with a cost lower than the threshold cost of that destination as well as the cost from source to that destination. Node f_1 is added to set of acknowledged forwarders. If all the destination nodes have been marked then it will stop the transmission else it will continue transmitting. Let f_2 be the next node to receive the complete message. It will send an ACK to *S*. *S* adds f_2 to the set of acknowledged forwarders and marks each such destination node which is reachable by f_2 with a cost lower than its threshold cost and the cost from source to that destination. It checks again if all the destinations have been marked. If so, it stops transmission else it continues transmitting and the whole process repeats until all destination nodes have been marked. The source then finds the forwarder, from the set of acknowledged forwarders, which can reach the maximum number of destinations with a cost lower than the corresponding threshold costs and the corresponding source to destination costs. All such destinations are assigned to the destination set of the forwarder and removed from the destination set of the source. If the destination set of the source does not become empty, it similarly finds again the forwarder which can reach the maximum number of destinations in the updated destination set i.e. the set of destinations which have not been removed yet. It assigns those destinations to that forwarder's destination set and removes them from the current destination set of the source. The process continues until the destination set of the source becomes empty. At the end of this process we will have a destination set for each forwarder. Now source directs each forwarder to begin forwarding to its corresponding destination set, if it is not empty. The transmission from any forwarder node to its own destination set will be same as the process just described with that forwarder acting as the source. Algorithm 3 presents the pseudo code for choosing the forwarders at a transmitting node *S*.

Algorithm 3: Pseudo Code for *MFS-RA*

Input: $D = \{d_1, d_2, \ldots, d_L\}$, $F = \{f_1, f_2, \ldots, f_J\}$

$C_{f_j,d_l}, C_{S,d_l} \forall j, l; \alpha$

1. $C_{th_l} = (1+\alpha) \times \min_j C_{f_j,d_l}$
2. Initialize: $D_{sub_j} = \phi \ \forall \ j = 1, 2, \ldots J; U = D; A_{fwd} = \phi$
3. Start message transmission using rateless approach.
4. until (ACK from any f_j)

   $A_{fwd} = A_{fwd} \cup f_j$

   $\forall \ d_l : C_{f_j,d_l} < C_{S,d_l} \ \& \ C_{f_j,d_l} \le C_{th_l}$

   $U = U - d_l$
5. If ($U == \phi$)

   Stop transmission

   Else

   Continue Transmission and go to 3.
6. Reinitialize $U = D$
7. $\forall \ j : f_j \in A_{fwd}$

   $D_{sub_j} = \{ d_l \in U : C_{f_j,d_l} < C_{S,d_l} \ \& \ C_{f_j,d_l} \le C_{th_l} \}$
8. $k = \underset{j: f_j \in A_{fwd}}{\operatorname{argmax}} | D_{sub_j} |$
9. Ask f_k to transmit to D_{sub_k}
10. $U = U - D_{sub_k}$
11. If ($U \ne \phi$)

    go to 7.

Chapter 4

SIMULATION RESULTS

We measured the performance of the algorithms proposed in the previous chapter on two test networks: Network I given in Fig. 4.1 and Network II given in Fig. 4.2 in terms of the mean number of transmissions per packet (MNT) and the mean transmission time (MTT). Mean number of transmissions per packet is defined as the mean number of packet transmissions required to send a message to all destinations divided by the number of packets per message. Mean transmission time is defined as the time taken on an average for a message to reach all destinations. Time taken to transmit one packet across a link is taken as the unit time.

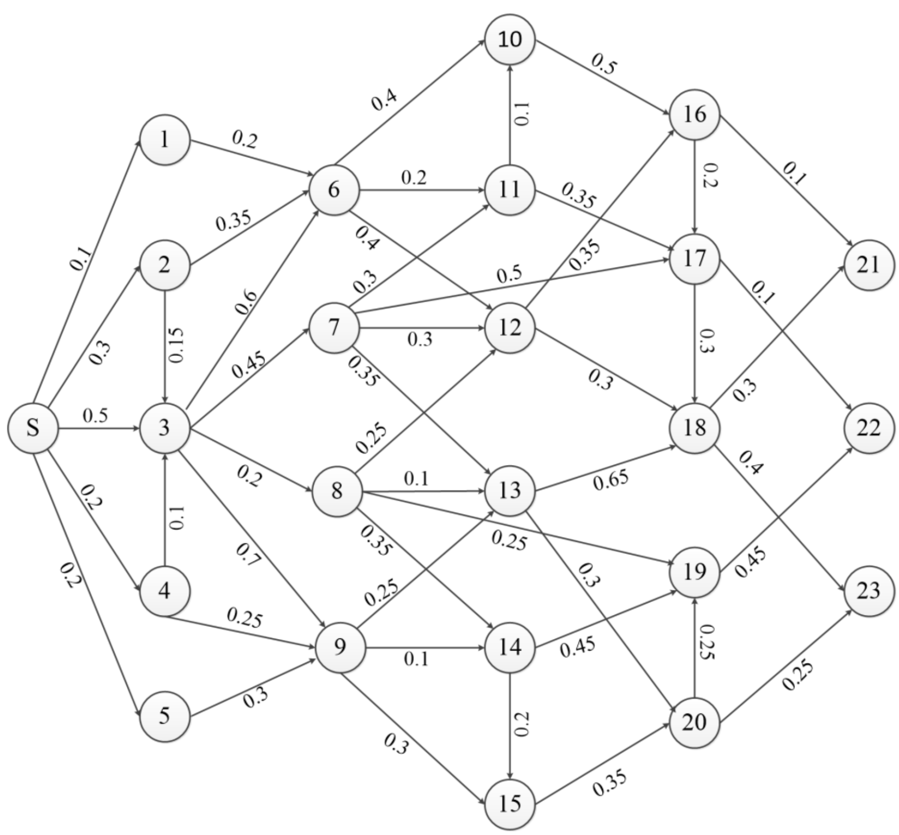

Fig. 4.1 Network I, link error probabilities are indicated next to links

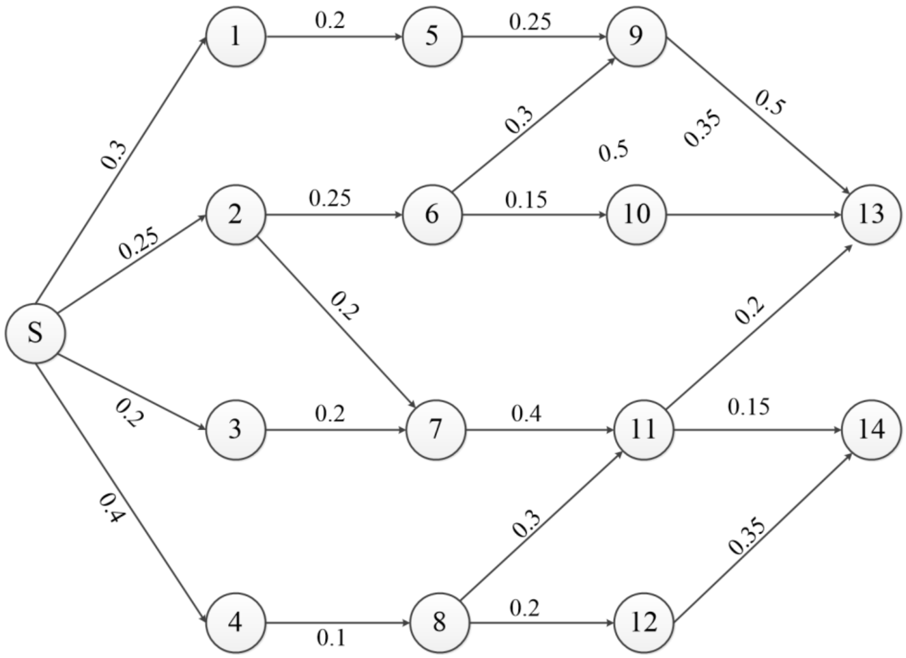

Fig. 4.2 Network II, link error probabilities are indicated next to links

4.1 Discussion on α

In *CFS-RA* and *MFS-RA* we studied the effect of parameter α by varying it for different sets of destinations in Network I and measured the performance. In Table 4.1 we have given the performance for the values of α which minimize the MTT in *CFS-RA* and MNT in *MFS-RA* respectively, for each of the various sets of destinations. We have compared this to the performance in the unrestricted case ($\alpha = \infty$). From the table it is clear that a proper selection of α can significantly improve the performance. In general, the optimal choice of α depends on the source, the set of destinations and the network. However, for *CFS-RA* it is observed that a small value of α, around 0.2, gives good performance.

Table 4.1 Performance of CFS-RA and MFS-RA in Network I for optimal α. Comparison is made with α = ∞ performance

Destination Set	***CFS-RA***					***MFS-RA***				
	α	*MTT*	*MNT*	$\alpha = \infty$ *MTT*	$\alpha = \infty$ *MNT*	α	*MTT*	*MNT*	$\alpha = \infty$ *MTT*	$\alpha = \infty$ *MNT*
1,14,21	0.1	283.3	9.77	330.6	10.98	2.9	337.0	8.42	338.3	8.45
16,18,20	0.2	230.7	10.58	296.1	14.98	0.1	230.4	10.54	307.1	12.50
16,22,23	0.2	293.4	11.41	428.4	13.44	0.1	332.9	10.20	456.9	11.42
8,15,17,21	0.1	283.2	13.82	329.8	14.22	2.0	336.3	9.70	337.4	9.74
3,7,10,19,23	0.2	292.3	12.32	428.8	17.78	0.5	396.3	11.64	379.0	12.1

For *CFS-RA*, a low value of α seems to perform well in general. This may be understood by looking at the two goals of *CFS-RA*, i.e. (a) Pass on the message to the forwarders as quickly as possible and (b) Do not deviate too much from the shortest path. A small value of α ensures we stay close to the shortest path. While this may slightly slow down the first goal in the first hop from the source, subsequent hops will be fast enough because the neighbor along the shortest path could be reached by a low erasure link and hence will be the first to send back an ACK, in general.

For *MFS-RA*, the value of optimal α seems fluctuating between low and high. The goal of *MFS-RA* is to minimize the branching as we go from the source to the destinations. If the destination set is such that a tree with a fewer number of branches from source to all destinations is possible, then a higher value of α which facilitates formation of such a tree is desirable. However if no such tree exists in the network, then it would be beneficial to keep α value low so that we at least do not meander away from the shortest path resulting in higher MNT.

4.2 MNT Performance

The performance of the three algorithms, on varying the number of packets per message, on the two test networks with the destination set being {8,15,17,21} in Network I and {7,9,12,13} in Network II in terms of MNT is presented in Table. 4.2 and Table. 4.3. The corresponding graphs are plotted in Fig. 4.4 and Fig. 4.5 respectively. We chose α = 0.1 for *CFS-RA* and α = 2.9 for *MFS-RA* for Network I. For Network II, α = 0.1 and 1.6 were chosen for *CFS-RA* and *MFS-RA* respectively.

Table 4.2 MNT Performance of three algorithms for Network I

Number of Packets per Message	***GFS-RA***	***CFS-RA***	***MFS-RA***
10	14.40	15.14	10.49
20	14.03	14.37	10.07
30	13.87	14.01	9.83
40	13.74	13.82	9.71
50	13.64	13.73	9.63
60	13.58	13.65	9.59
70	13.52	13.61	9.54
80	13.50	13.58	9.51
90	13.49	13.52	9.47

Table 4.3 MNT Performance of three algorithms for Network. II

Number of Packets per Message	***GFS-RA***	***CFS-RA***	***MFS-RA***
10	11.07	10.43	9.72
20	10.77	9.92	9.02
30	10.62	9.81	8.79
40	10.57	9.74	8.60
50	10.43	9.70	8.55
60	10.37	9.66	8.50
70	10.32	9.60	8.44
80	10.27	9.54	8.41
90	10.25	9.51	8.37

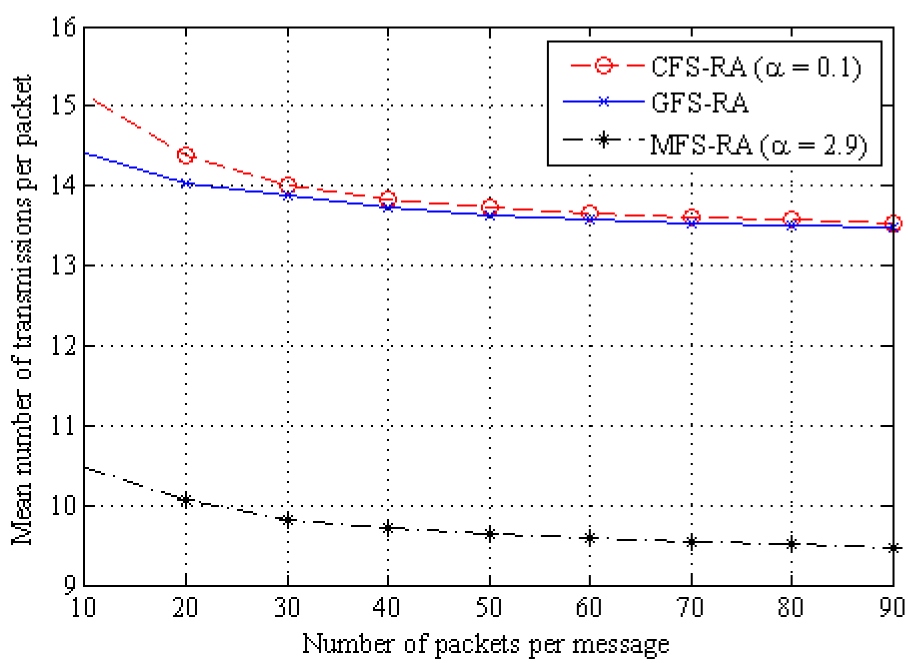

Fig. 4.4 Mean Number of Transmissions per packet (MNT) Performance of all the algorithms as a function of number of packets per message on Network I, destination set = {8.15,17,21}

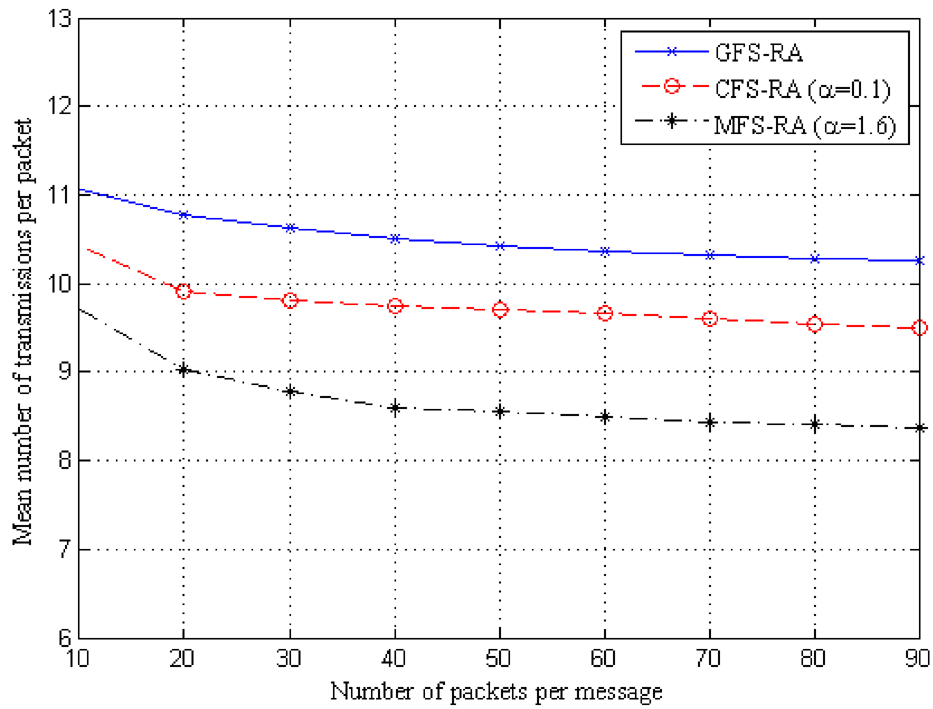

Fig. 4.5 Mean Number of Transmissions per packet (MNT) Performance of all the algorithms as a function of number of packets per message on Network II, destination set = {7,9,12,13}

In terms of MNT, *MFS-RA*, as expected, performs much better than the other two algorithms. The general nature of the MNT plot as a decreasing function of number of packets in a message can be explained by the fact that the link cost as defined by (2.7), for a fixed value of p, decreases as N increases beyond a small value as depicted in Fig. 4.6.

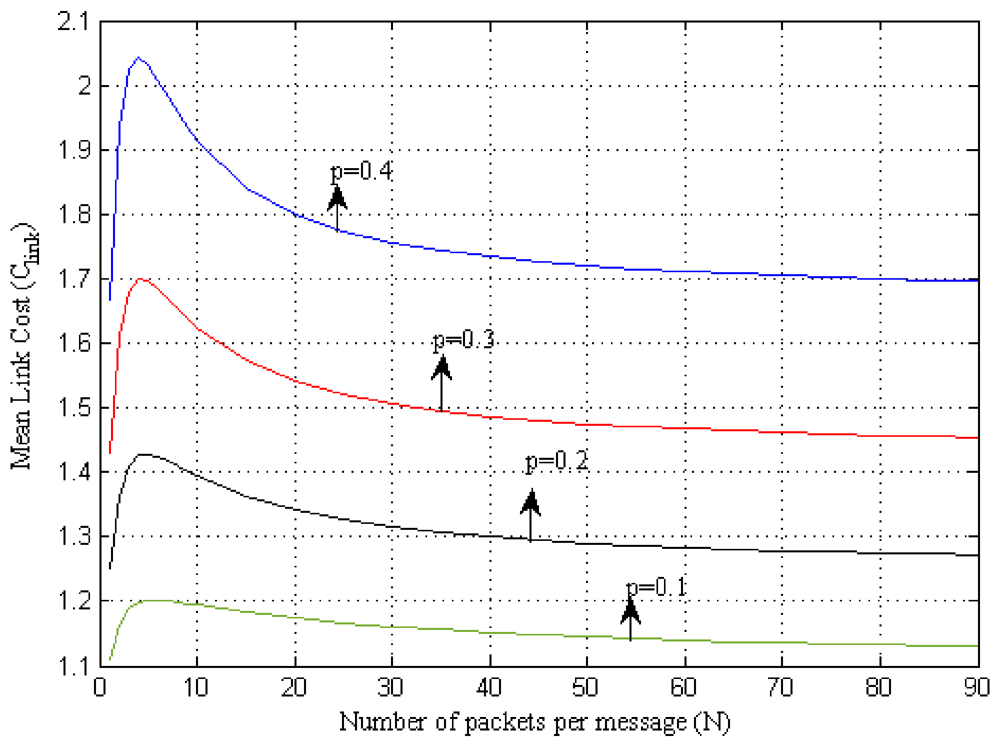

Fig. 4.6 Variation of mean link cost (C_{link}) with N for different erasure probabilities *p*

4.3 MTT Performance

The performance of the three algorithms, on varying the number of packets per message, on the two test networks with the destination set being {8,15,17,21} in Network I and {7,9,12,13} in Network II in terms of MTT is presented in Table. IV and Table. V. The corresponding graphs are plotted in Fig. 4.7 and Fig. 4.8 respectively. We chose $\alpha = 0.1$ for *CFS-RA* and $\alpha = 2.9$ for *MFS-RA* for Network I. For Network II, $\alpha = 0.1$ and 1.6 were chosen for *CFS-RA* and *MFS-RA* respectively.

Table 4.4 MTT Performance of three algorithms for Network I

Number of Packets per Message	***GFS-RA***	***CFS-RA* ()**	***MFS-RA***
10	92.7	76.1	90.9
20	181.3	146.1	174.5
30	266.8	215.0	255.2
40	346.9	283.2	336.6
50	428.4	351.8	416.9
60	507.3	419.8	498.7
70	587.7	487.6	578.9
80	668.8	556.1	659.6
90	748.3	623.7	737.9

Table 4.5 MNT Performance of three algorithms for Network II

Number of Packets per Message	***GFS-RA***	***CFS-RA***	***MFS-RA***
10	63.9	54.9	63.1
20	121.3	105.9	118.4
30	177.6	156.3	173.5
40	234.5	205.9	226.9
50	292.1	256.2	282.8
60	349.2	305.2	337.8
70	407.1	354.9	391.3
80	464.6	404.4	446.6
90	523.1	454.3	499.3

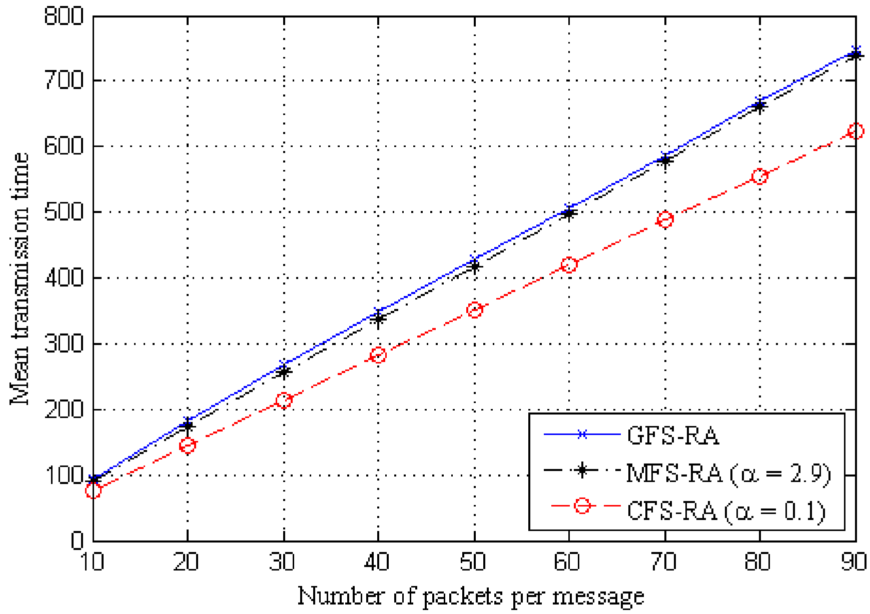

Fig. 4.7 Mean Transmissions Time (MTT) Performance of all the algorithms as a function of number of packets per message on Network I, destination set = {8,15,17,21}

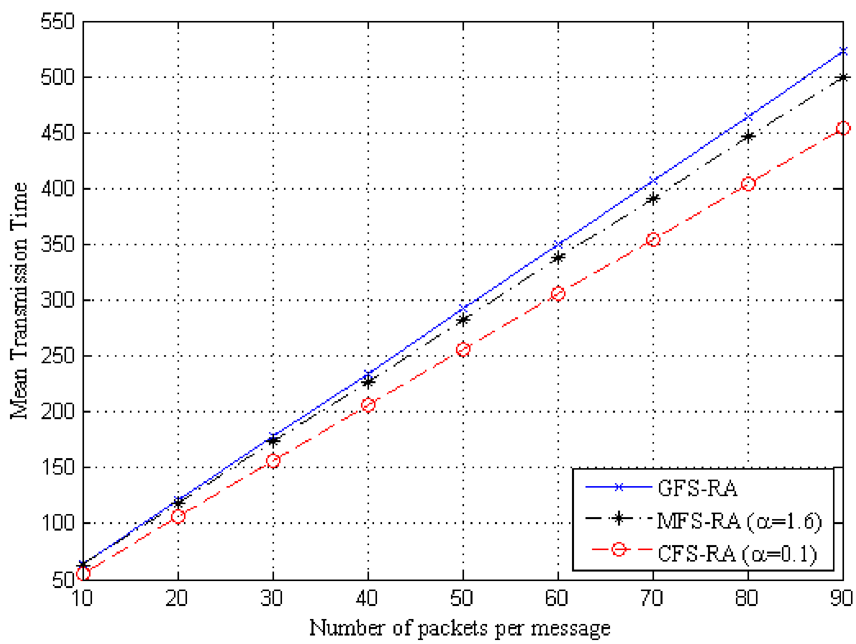

Fig. 4.8 Mean Transmissions Time (MTT) Performance of all the algorithms as a function of number of packets per message on Network II, destination set = {7,9,12,13}

It can be seen that in terms of MTT, *CFS-RA* outperforms the other algorithms substantially.

Chapter 5

CONCLUSIONS & FUTURE DIRECTIONS

5.1 Conclusions

The work proposed algorithms for multicasting in an opportunistic wireless network using rateless codes. Three opportunistic routing algorithms were proposed in total. *GFS-RA* is a basic greedy algorithm which offers relatively poor performance and hence is not recommended for use. *CFS-RA* is an algorithm which seeks to reduce the forwarding cost by choosing a forwarder as quickly as possible without veering away too much from the shortest route that offers great performance in terms of mean transmission time for message delivery, in spite of having a relatively higher mean number of transmissions per packet. *MFS-RA*, which greedily chooses the forwarders so as to form a multicast tree with the minimum number of branches, is the best algorithm in terms of minimizing mean number of transmissions done per packet, though its performance in terms of the mean time taken for transmission is not as good as *CFS-RA*. Hence, we recommend *CFS-RA* for applications requiring low network delay or latency. On the other hand, *MFS-RA* is recommended for energy efficient applications requiring transmissions to be done judiciously.

5.2 Future Directions

Future work may include:

a) **Multicast cost metrics**: The present work uses unicast cost metrics for making routing decisions. Instead, multicast cost metrics that describes the cost of multicasting from a single node to a set of destinations could be proposed. The disadvantage with such a cost metric is their high complexity which make them infeasible to be used in a practical system. So if possible, an elegant and simple multicast cost metric could be created which might enable the creation of efficient multicast routing protocols with performance that could be theoretically established and guaranteed.

b) **Using other coding schemes**: This work uses a simple abstraction of rateless coding scheme for transmission. Instead some other coding scheme, rateless or non-rateless like network coding could be used so as to achieve a better performance. Also the proposed algorithms themselves are quite independent of the rateless coding scheme used and could apply to many other coding schemes by suitably changing the cost metric. In such cases, the performance of the algorithms under a general coding scheme could be assessed.

BIBLIOGRAPHY

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, New York, 1990.

[2] Sohn, K., Kim, H. S., & Kim, Y. Y. (2013), "A novel multicast scheme for feedback-based multicast services over wireless networks," *EURASIP Journal on Wireless Communications and Networking*, 2013(52), 1–12.

[3] Luby, M., "LT codes," *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on* , vol., no., pp.271,280, 2002

[4] MacKay, D.J.C., "Fountain codes," *Communications, IEEE Proceedings-* , vol.152, no.6, pp.1062,1068, 9 Dec. 2005

[5] Byers, J.W.; Luby, M.; Mitzenmacher, M., "A digital fountain approach to asynchronous reliable multicast," *Selected Areas in Communications, IEEE Journal on* , vol.20, no.8, pp.1528,1540, Oct 2002 doi: 10.1109/JSAC.2002.803996

[6] R. R. Choudhury and N. H. Vaidya, "MAC-layer anycasting in ad hoc networks," *SIGCOMM Comput. Commun. Rev*., vol. 34, no. 1, pp. 75–80, 2004.

[7] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," *in Proc. ACM SIGCOMM*, Philadelphia, PA, 2005, pp. 133–144.

[8] Indela, S.; Mishra, S.K.; Bose, S.K.; Wen-De Zhong, "Network routing over erasure channels using systematic rateless codes," *TENCON 2014 - 2014 IEEE Region 10 Conference* , vol., no., pp.1,6, 22-25 Oct. 2014

[9] H. Dubois-Ferriere, M. Grossglauer, and M. Vetterli, "Least-cost opportunistic routing," in *Proc. 2007 Allerton Conf. Commun., Control, Comput.*, pp. 994–1001

[10] Dubois-Ferriѐe, H.; Grossglauser, Matthias; Vetterli, M., "Valuable Detours: Least-Cost Anypath Routing," *Networking, IEEE/ACM Transactions on* , vol.19, no.2, pp.333,346, April 2011

[11] Mazumdar, S.P.; Bose, S.K.; Wen-De Zhong, "Multicast Least Cost Anypath Routing: Route Cost Calculations and Forwarding," *Communications Letters, IEEE* , vol.16, no.10, pp.1652,1655, October 2012

[12] Subramanian, V.G.; Leith, D.J., "On a class of optimal rateless codes," *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on* , vol., no., pp.418,425, 23-26 Sept. 2008

[13] Wu, C.W.; Tay, Y.C., "AMRIS: a multicast protocol for ad hoc wireless networks," *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE* , vol.1, no., pp.25,29 vol.1, 1999 doi: 10.1109/MILCOM.1999.822636