ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

"ΣΧΕΔΙΑΣΜΟΣ ΑΛΓΟΡΙΘΜΟΥ ΕΚΠΟΜΠΗΣ ΣΕ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΧΑΜΗΛΗΣ ΚΑΤΑΝΑΛΩΣΗΣ"

"Design of a broadcasting algorithm in wireless low-power networks"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΝΙΚΟΛΑΟΥ ΚΙΑΜΙΛΗ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ :
ΑΡΓΥΡΙΟΥ ΑΝΤΩΝΙΟΣ

ΣΥΝΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:
ΚΟΡΑΚΗΣ ΑΘΑΝΑΣΙΟΣ

Στην οικογένεια μου και τους φίλους μου,

CONTENTS

ΠΕΡΙΛΗΨΗ

Στην διπλωματική αυτή παρουσιάζουμε ένα νέο αλγόριθμο μέτρησης της ποιότητας των ασύρματων links με στόχο την ελάττωση της ενεργειακής κατανάλωσης και της επιβάρυνσης στο throughput του δικτύου με παράλληλη διατήρηση των ποσοστών ευστοχίας της εκάστοτε εκτίμησης σε υψηλά επίπεδα. Ως βάση αυτού του αλγορίθμου χρησιμοποιούμε τον ETX και δοκιμάζουμε την λειτουργικότητα του δικού μας αλγορίθμου ονόματι RC-ETX σε σχέση με τον προαναφερθείσα πάνω σ'ένα ευκαιριακό δίκτυο στο οποίο εφαρμόζεται ο Multipath Opportunistic Routing Engine (MORE).

Στις 3 πρώτες ενότητες της διπλωματικής γίνεται μια περιγραφή των αλγορίθμων MORE και του ETX καθώς και η αξιολόγηση των προτερημάτων και αδυναμιών του τελευταίου. Επίσης εξηγείται ο τρόπος λειτουργίας αυτών καθώς και τον τρόπο με τον οποίο ο MORE στηρίζεται στην ETX μετρική.

Στην 4η ενότητα παρουσιάζουμε τον δικό μας αλγόριθμο εξηγώντας τις ανάγκες που μας οδήγησαν στην ανάπτυξη του και περιγράφοντας τον τρόπο λειτουργίας του. Έπειτα παρατίθενται δύο πιθανά σενάρια λειτουργίας του όπου εξηγούνται σταδιακά τα βήματα του αλγορίθμου. Επιπλέον παρουσιάζονται και ερμηνεύονται τα αποτελέσματα της προσομοίωσης που δημιουργήθηκε για να συγκριθεί ο αλγόριθμος μας με αυτόν του ETX. Τέλος γίνεται μία αξιολόγηση της μετρικής και αναλύονται τα μειονεκτήματα και τα πλεονεκτήματα της.

Κλείνοντας στη 5η ενότητα περιγράφονται μελλοντικοί στόχοι και πιθανές βελτιώσεις του αλγορίθμου σε συγκεκριμένους τομείς όπου υστερεί.

ABSTRACT

In this thesis we propose a new algorithm for estimating the delivery ratio of wireless links in order to reduce power consumption and the total algorithm's overhead on the data transmission process while maintaining high accuracy expectation values. The ETX metric algorithm is used as the basis of our creation and is also the algorithm that ours, named RC-ETX, will compete with in terms of functionality on opportunistic networks and more specifically networks running MORE.

The three first chapters of our thesis include the descriptions of MORE and ETX algorithms and then we evaluate the advantages and the disadvantages of the latter. In these chapters it is also explained how do they work and how does MORE utilize the ETX metric.

In chapter four we introduce our algorithm named RC-ETX while explain its importance and how does it work. Afterwards we examine two possible and distinct scenarios were each step of the algorithm is shown distinctively in order to get a better grasp on our idea. In this chapter we also show the results of the simulation we have created and we try to interpret them accordingly. Then we perform a comparison between ETX and RC-ETX and finally we evaluate our metric and discuss its pros and cons.

Concluding on chapter 5 we describe ways to improve our proposed algorithm and suggest some improvements in certain aspects of the algorithm in order to improve it and make it more reliable.

1. INTRODUCTION

In modern days wireless networks have become an irreplaceable part of our everyday life. From wireless telephony and home networks to radars and space expedition communication is achieved through the wireless medium using different hardware and communication protocols. Given this widespread utility of this medium, there have been numerous and continuous researches to improve the overall transmission quality while certain constraints are applied, through new software and hardware techniques.

One of the most famous results of these researches was the creation of the IEEE 802.11 standard commonly known by its marketing name as Wi-Fi. 802.11 is a collection of protocols which describe the implementation of wireless local area networks (WLANs). Such networks operate at 2.4, 3.6, 5 and more recently 60 Hz frequency bands and are the most commonly used along with the GSM/UMTS networks which are used in wireless telephony. Throughout its short history the 802.11 standards set has been optimized in different perspectives such as power consumption, bandwidth, latency mitigation, QoS improvement etc. To achieve such optimizations new algorithms and techniques have been developed to help with the new implementations of the standard. Some examples of such algorithms are: the ETX and ETT link metrics which are used to approximate the actual link loss probabilities, Back-Pressure and Enhanced-BP algorithms who achieve maximum bandwidth on a large scale network, SRCR algorithm that targets to achieve low data transfer delays etc.

One of the most emerging trends of today's research on wireless networks is the utilization of opportunistic overhearing of packets in order to improve the overall network's bandwidth. Packet overhearing is commonly encountered in broadcast transmissions.

Many algorithms have been implemented that take advantage of such communications between wireless nodes with the most prominent being the MORE (Multi-path Opportunistic Routing Engine). MORE utilizes the ETX metric in order to calculate link loss probabilities and then uses them in conjunction with a predetermined forwarder list. Also it features random linear network coding to improve overall throughput.

Although been a powerful algorithm in general its power consumption does not allow it to easily be applied on systems that run on batteries. One of the causes of this side back is the frequent use of the ETX metric.

With all the above being said in this thesis we try to address the above issue by developing a new metric which achieves almost the same results as ETX but with lower power consumption specifically targeted to wireless networks that use random linear coding to transmit data (regardless of being opportunistic based or unicast based). We introduce the RC-ETX (reduced consumption ETX) a new metric that is based on ETX.

RC-ETX also counts the number of retransmissions of a packet required in order for a node to hear a single packet destined for him. The difference with the ETX metric is that in RC-ETX we limit the probe packets sent and switch from time-based metric calculation frequency to batch-based frequency.

Finally I perform the tests on a software simulator which runs the MORE algorithm with ETX support and one with RC-ETX. The test results show that the actual link accuracy is close to that of ETX but the consumption of RC-ETX is significantly lower than that of simple ETX.

2. DESCRIPTION OF THE MORE ALGORITHM

2.1 INTRODUCTION TO MORE

MORE is a broadcast based algorithm. Each node transmits a packet by broadcasting it to the wireless medium allowing reception by any node that exists in the network unlike traditional routing protocols which determine the next hop of a packet just before its transmission. Broadcast transmissions allow a packet to be received by a different subset of nodes [citations on MORE paper intro] each time due to the wireless medium dynamic characteristics

.

Traditional routing does not allow such transmissions to be made mostly because they're unicast based protocols that forward the packet hop by hop on a predetermined route. On the other hand opportunistic routing allows packet overhearing hence exploiting fortunate receptions by nearby nodes creating new multipath routes towards the destination in order to increase the throughput. Although throughput increases so does the number of packet transmissions; nodes that heard a certain packet will retransmit it as soon as they get medium access wasting wireless resources and reducing any potential throughput gain.

MORE allows nodes to forward their packets without having to coordinate with each other by building on the theory of network coding.

2.2 EXPLANATION OF TERMINOLOGY

2.2.1 OPPORTUNISTIC ROUTING

As said before MORE is an opportunistic routing algorithm. What is opportunistic routing?

Opportunistic routing is routing based on packets being broadcast over the network without the need of a route from the source. After a broadcasted packet is being received by an intermediate node the node that is part of the best route to destination is selected as the next forwarder. The gains from this approach are two-fold. First, the node that receives a packet has many chances to be closer to the destination (in terms of reception probability) and second is the throughput gain caused by the ability to use low-quality links which are not used in traditional routing.

2.2.2 NETWORK CODING

When a node utilizes network coding it actually transmits new distinct packets which are generated by random linear combinations of the packets they received without the need to coordinate which node forwards which packets.

2.2.3 MORE DEFINITIONS USED

These terms described here are used in this thesis as they're found on the original MORE paper.

1) *Native Packet:* A native packet is a non-coded packet usually existing only on the source. Native packets are not transmitted; first the node combines them and then transmits them.

2) *Batch:* Batches (or "encode buddies") are groups of native packets which are coded together. Each batch may or may not be the same from one to another. Native packets belonging to one batch cannot be combined with native packets from another batch.

3) *Coded Packet:* A coded packet is a linear combination of native packets belonging to the same batch. Encoding is achieved by multiplying each byte of a native packet by a number c and then adding the two packets together by XORing the corresponding bytes in each packet. In the case we have more than 2 packets per batch we simply multiply each one of them by a random coefficient c_i and then XOR all the packets together. Thus an encoded packet is produced. Also coding previously linearly combined packets with each other results in linear combinations of the native packets themselves.

4) *Code Vector of a Packet:* Code Vector as its name implies is a vector which contains all the coefficients of the native packets combined on a single coded packet. It is used to describe how to derive the coded packet from the native packets. A coded packet can be a native packet in case its code vector has the element c_i value equal to 1 and the rest elements are 0's.

5) *Linearly independent Packets:* Two or more coded packets are linearly independent when their corresponding code vectors are linearly independent too.

6) *Innovative Packet:* An innovative packet is a packet that is linearly independent from the previous packets the node has received. Such packets contain new useful information as opposed to the linearly dependent packets which can have their information extracted from previously received packets. Non-innovative packets can be discarded safely.

7) *Closeness to Destination:* Closeness of a node x to destination is not the physical distance between these two nodes. Closeness to destination is determined by the number of transmissions required to deliver a packet from the node x to the destination. MORE utilizes ETX to estimate the aforementioned distance and uses periodic pings also known as ETX probe packets to measure the average delivery probability between a pair of nodes.

8) *Downstream/Upstream:* A node x is downstream of another node y when x is closer to the destination than node y. considering this, node y is also upstream of node x.

2.3 THE MORE ALGORITHM

2.3.1 HOW DOES IT WORK

MORE is an algorithm designed for stationary multi-hop wireless networks and community wireless networks where nodes are not limited by processing power or by memory. In order to simplify the way MORE works we separate the nodes into 3 categories: senders, forwarders and destinations.

1) The sender:

A sender is usually the source of the information which sends batches of packets as described before. The source creates a linear combination of the packets included in the current batch when the 802.11 MAC permits and then broadcasts the coded packet. Each coded packet contains a coded vector that describes its contents, its current batch number and the distance of the transmitter to the destination. The source continues to send random linear combinations of packets from a certain batch until it receives an ACK from the destination that it has received a number of innovative packets equal to the number of the packets in the current batch which is being transmitted. ACK utility and implementation is described later in this section.

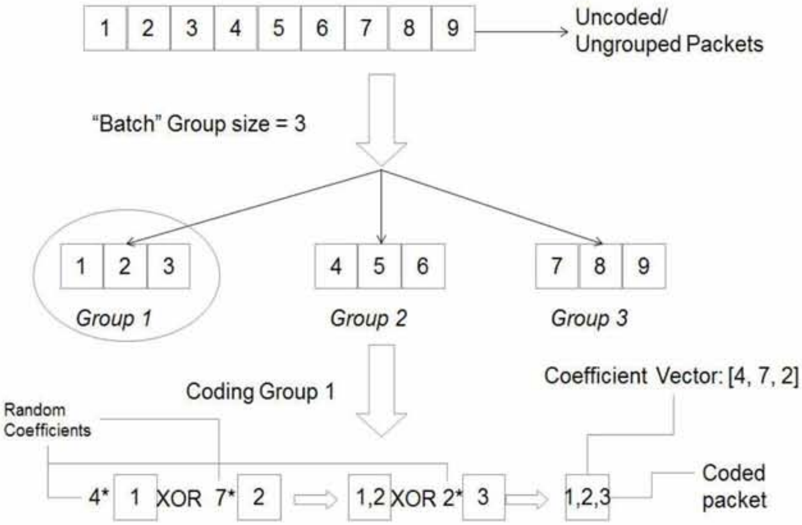


Figure 1. The sender's encoding sequence

2) The forwarder:

Forwarders are a group of nodes that listen to all transmission being made but are not the destination of the packets being heard. When a forwarder receives a packet, it checks whether it is an innovative one. Innovation check can be done by a simple Gaussian elimination performed on the code vectors of the coded packets received by the node. The packet may be discarded if it's found being not innovative or may be buffered by the node along with the previously received packets from the same batch. Afterwards the node checks whether it is closer to the destination than the sender of the coded packet. If this is true then the node broadcasts a new linear combination of the packets from the same batch as the recently received packet.

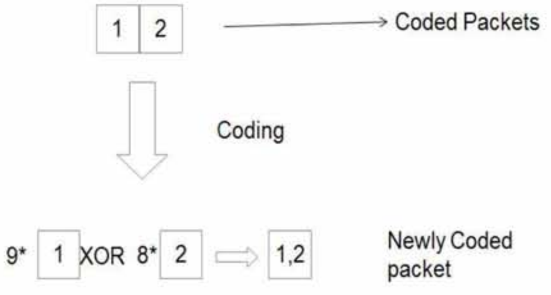


Figure 2. The coding process on each forwarder

3) The destination:

The destination checks whether a packet it receives is innovative. If it isn't then it's discarded. If it's not then it is buffered along the previously received coded packets from the same batch as the newly arrived packet. When the node has a number of innovative packets equal to the number of the packets included in the same batch then it can decode the original packets using simple matrix inversion. The destination also sends an ACK when it has the sufficient number of innovative packets through the shorted route towards the source in order prevent it stop sending packets of the current batch and move to the next batch. Nodes that have packets from the previous batch will have them dropped when they reach their batch timeout limit or when packet with a higher sequence number is received.

$$\begin{pmatrix} p_1 \\ \vdots \\ p_K \end{pmatrix} = \begin{pmatrix} c_{11} & \dots & c_{1K} \\ \vdots & \ddots & \\ c_{K1} & \dots & c_{KK} \end{pmatrix}^{-1} \begin{pmatrix} p'_1 \\ \vdots \\ p'_K \end{pmatrix}$$

Figure 3. The process of decoding. Where p symbolizes the native packets p' the encoded ones and c are each encoded packet's code vectors

2.3.2 EXTRA FEATURES

As a robust network coding algorithm MORE has to overcome several issues in order to be implemented successfully. One of the most common causes of failure of this algorithm is the fact that it codes all the packets it has overheard from the current batch. Combining all these packets which number might be overwhelming can lead to CPU bottlenecks. In order to avoid such a misfortune MORE employs three different techniques:

1) Code only Innovative Packets:

Since coding every packet received increases the coding cost and since coding non-innovative packets does not produce or add any useful information into the newly created coded packet it is safe to discard such packets. By discarding non-innovative packets we reduce the amount of memory needed to store all of the overheard packets and we limit it up to a maximum equal to the number of the packets per batch. Moreover we reduce the CPU overhead by simply combining only innovative packets to create new coded packets.

2) Operate on Code Vectors:

Operations such as Gaussian elimination to determine whether a received packet is innovative require a lot of processing power. That's why MORE operates solely on code vectors instead of operating on all the combined data limiting the operations performed on a packet's data. This means that each packet's data are touched only when we create a new coded packet and not during the innovation check.

3) Pre-coding:

Pre-coding is the process of creating a coded packet using the data already existing in the buffer while the node does not have access to the medium. Instead of combining all the packets when the MAC allows access to the medium or creating many of them and storing them to the output queue while waiting for a MAC permission. Also if a new packet arrives just before a coded packet is transmitted then it is multiplied by a random coefficient and added to the pre-coded packet. This results in less time consumption than generating a new coded packet from scratch each time a new packet is received.

Another problem that arose was the numbed of the data each node would forward. If a node x has a lower delivery probability to the destination than the node y then node x should forward less data than node y since forwarding the same number of packets as y would waste a lot of the network's bandwidth on failed transmissions. Therefore node x should only supplement only the extra information that has not been heard by node y.

In order to address this issue the following algorithm is used:

i) Let N be the number of nodes in the network where node 0 is the destination and node N-1 is the destination.
ii) For two nodes a, b let a<b denote that node a is closer to the destination than node b in terms of ETX magnitude. The higher the ETX the greater the distance.
iii) Let e_{ij} denote the loss probability when node i sends a packet to node j
iv) Let z_i denote the expected number of transmissions required for the node i to route a packet to the destination successfully
v) Let L_i denote the expected number of packets that node i must forward

The expected number of packets that node j receives from nodes with higher ETX is:

$$\sum_{i>j} z_i(1-e_{ij})$$

The L_j of this node should be:

$$\sum_{i>j} (z_i(1-e_{ij})\prod_{k<j} e_{ik}) \quad (1)$$

Note that $\prod_{k<j} e_{ik}$ denotes the forwarding probability when the node j receives a packet and no other node with ETX lower than that of j's has the packet.

Since the source generates the packet we have:

$$L_s = 1$$

The Eq. (1) shows us the number of packets j has to forward. Knowing this, the expected number of transmissions that j must make is:

$$z_j = \frac{L_j}{(1 - \prod_{k<j} \epsilon_{jk})}$$

The algorithm that calculates the above z for each node i is briefly described below:

```
for $i = n \ldots 1$ do
    $L_i \leftarrow 0$
$L_n \leftarrow 1$    {at source}
for $i = n \ldots 2$ do
    $z_i \leftarrow L_i / (1 - \prod_{j<i} \epsilon_{ij})$
    $P \leftarrow 1$
    for $j = 2 \ldots i - 1$ do
        {compute the contribution of $i$ to $L_j$}
        $P \leftarrow P \times \epsilon_{i(j-1)}$    {here, $P$ is $\prod_{k<j} \epsilon_{ik}$}
        $L_j \leftarrow L_j + z_i \times P \times (1 - \epsilon_{ij})$
```

Where node n is the source and node 1 is the destination.

2.4 SIDEBACKS

As robust as it may seem MORE has several issues due to its centralized approach making it difficult to be applied on realistic networks where communication between nodes might require more than one or two hops.

That being said MORE's approach is not sufficient on distributed systems and requires a special scheduler who dictates each node when to transmit. This special scheduler is able to calculate a metric called TX_credit and pass it to each node successfully. The TX_credit of a node is the number of transmissions that a node should make for each packet it receives from a node with a higher ETX metric to destination than its own. This means that the TX_credit metric is used only if the sender is farther to the destination than the receiving node.

The TX_credit of node i is calculated as it follows:

$$\text{TX\_credit}_i = \frac{z_i}{\sum_{j>i} z_j (1 - \epsilon_{ji})}$$

This concludes the description of the MORE Algorithm. In my thesis I use MORE as described above as the basis of my simulation.

3. DESCRIPTION OF THE ETX METRIC

3.1 INTRODUCTION

Expected Transmission Time (ETX) is a metric first described on a paper written by Douglas S. J. De Couto, Daniel Aguayo, John Bicket and Robert Morris. According to its authors the ETX metric is capable of finding high-throughput paths on multi-hope wireless networks by minimizing the expected total number of packet transmissions required to deliver one packet from its source to the destination. It also incorporates the effects of asymmetric link loss between two nodes unlike previous methods like minimum hop count which chooses the path to the destination by taking in account all the possible routes to destination counts their total hop number from source to destination and then selects the path with the minimum hop count.

ETX's capabilities are shown on its implementation as a metric for the DSDV[12] and DSR[13] routing protocols where it outperforms the default metric previously used in these two algorithms and increases their overall throughput.

3.2 THE MAIN IDEA

In their research the authors of the paper note that a good link loss metric should be able to address 3 basic issues:

i) The wide range of link loss ratios
ii) The asymmetric nature of links who have different uplink and downlink loss ratios
iii) The interferences between successive hops which exist on multi-hop paths which is pretty common on wireless networks.

Many previous approaches to solve the aforementioned issues such as minimum hop count and the per-link delivery ratios have fallen short to address all of them at once. The first one does not guarantee that a destination is reachable through the shortest hop count path and the second one does not calculate the inter-hop interference that exists when the transmissions take place. ETX does in fact address all the above issues successfully.

3.3 THE METRIC

The ETX metric of a link is actually the number of retransmissions required to send a packet from one end of the link to the other. It utilizes the forward and reverse delivery ratios of a link. The forward delivery ratio d_f of a node Y is the number of packets received by a node X divided by the total number of packets sent by node Y. The reverse delivery ratio d_r is calculated on node X and is the probability that the ACKs sent from Y will be successfully received by node X. This being said the reverse probability ratio is the number of ACKs received by node X divided by the total number of packets X has transmitted.

$$d_r = \frac{ACKS_{received}}{ACKS_{expected}}$$

$$d_f = \frac{packets_{heard}}{counted\_packets}$$

The multiplication of d_f by d_r denotes the probability of a data packet being successfully received and acknowledged. If a data packet is not acknowledged will be retransmitted by its source. Due to the fact that each attempt to transmit a packet is considered a Bernoulli trial the writers of the paper conclude that the expected number of transmissions for a link is approximated as:

$$ETX = \frac{1}{d_r * d_f} \quad (2)$$

In order for this equation to be valid it is assumed that the reverse and forward delivery probabilities are constant for a given link or for the duration of link measurements.

3.4 THE MEASUREMENT PROCEDURE

The algorithmic implementation of the ETX metric is based on the theory described above. More specifically the metric is obtained by each node using probe packets which are transmitted periodically. We define the time period when ETX is being measured as ETX period and the frequency this period occurs is called ETX frequency. Each time an ETX period begins each node starts transmitting probe packets. Probe packets are data packets which include the number of probe packets received by the sender from each of its neighbors. Sending probe packets like this gives the receiver of a probe packet the capability to calculate the forward delivery ratio to the sender. The reverse delivery probability is calculated on the sender by dividing the ACKs it heard by the total number of probe packets sent to each of its neighbors. When the ETX period ends, each node calculates the expected number of retransmissions of its links using the simple equation shown in (2).

When decisions need to be made across a multi-hop data route the total ETX for each link is the summation of the ETX metrics of every possible link that is exists between intermediate nodes

4. THE RC-ETX METRIC

4.1 INTRODUCTION

As we can see although ETX is a very solid and complete link loss estimation metric it has some unnoticed drawbacks. For example ETX applied on an opportunistic network takes in account the reverse link loss probability which is the probability of an ACK transmission being lost. Such transmissions do not occur in opportunistic networks since data packets are broadcasted through the medium and broadcast transmissions do not send physical layer ACKs. One more drawback of the ETX metric is the fact that it is an energy deficient metric. It includes many transmissions of probe packets for extended periods of time and if it's used in conjunction with a routing algorithm like MORE which sends ETX probes periodically means that the energy requirements are increasing as data transfers continue. And while energy consumption is not a matter to worry about on nodes connected to a voltage source, it is of great importance on wireless devices that run on batteries such as mobile phones and wireless sensors.

This need for a low consumption and equally accurate metric as ETX is the leading force of our attempt to develop an algorithm for opportunistic networks that will perform as well as ETX while improving the aspect of energy consumption on the wireless nodes.

To complement the above need we have developed a new algorithm called RC-ETX. RC-ETX is heavily inspired by ETX as its name suggests in terms of the metric's interpretation but its approach is totally different. It mainly builds on the main characteristic of opportunistic networks which is the grouping of data packets into batches.

In its simulation RC-ETX is applied on MORE algorithm as a substitute for ETX where it performs actually well and on par with ETX on a static (no mobile nodes) network. Due to its nature RC-ETX is specifically designed for opportunistic networks and might not work as expected on other types of networks for reasons that will be described later on this chapter.

4.2 DESCRIPTION OF THE RC-ETX ALGORITHM

4.2.1 MATHEMATIC TERMS

RC-ETX procedure is based on the fact that on opportunistic networks there's nothing such as physical layer ACKs so it's safe to assume that since no ACKs are sent, the reverse delivery ratio of a link d_r is equal to 1. So according to the ETX metric that is:

$$ETX = \frac{1}{\mathrm{dr}*\mathrm{df}} \quad (1)$$

Therefore RC-ETX is equal to

$$RCETX = \frac{1}{\mathrm{d_f}} \quad (2)$$

In other word RC-ETX is the number of transmissions required for a packet to be successfully delivered from a node X to a node Y without the need of X receiving an ACK from Y which embodies the way opportunistic networks work.

4.2.2 FUNCTIONALITY

Before describing the procedure that our algorithm follows we must first introduce two specific changes to the core of the MORE algorithm:

1) We stop performing ETX measurements periodically in order to use our metric instead.
2) We implement two types of counters to each node of the network.
1. A counter for every different node Y a node X has received a packet from which is used to count the number of received packets from each node.
2. A counter that counts the number of packets a node has transmitted.

The algorithm has 3 steps, the first being the pre-emptive ETX. In this step an ETX session is being performed to calculate the link loss probabilities before the initiation of the data transmissions. The second step is the data transmission step. During this step the source transmits coded packets of a certain batch using the link loss probabilities calculated in the previous step. When the destination has received all the required linearly independent coded packets it is ready to decode the packets of the batch that is being transmitted. This is where the third and final step of the algorithm occurs. The third step is the probe packet transmission step. The destination initiates the procedure by sending one probe packet to each of its neighbors. The probe packet contains the number of packets a node X has heard from a certain node Y. Its neighbors do the same thing until every node has sent a probe to each of its neighbors once without retransmissions.

When a node receives a probe packet it checks whether it's the first it hears or not. If it's the first then it starts sending probes to its neighbors too. If it's not the first one it means that the aforementioned procedure has been initiated and it takes no action. After this check each node calculates its RC-ETX metric according to equation (2).

4.2.3 SCHEMATIC REPRESANTATION

Let's make the hypothesis that we have the following network where next to each link is the current estimated ETX link delivery probability. This probability occurs like this:

$$LP_{ij} = \frac{1}{ETX}$$

Where LP_{ij} is the estimated link delivery probability for node i to send a packet successfully to node j.

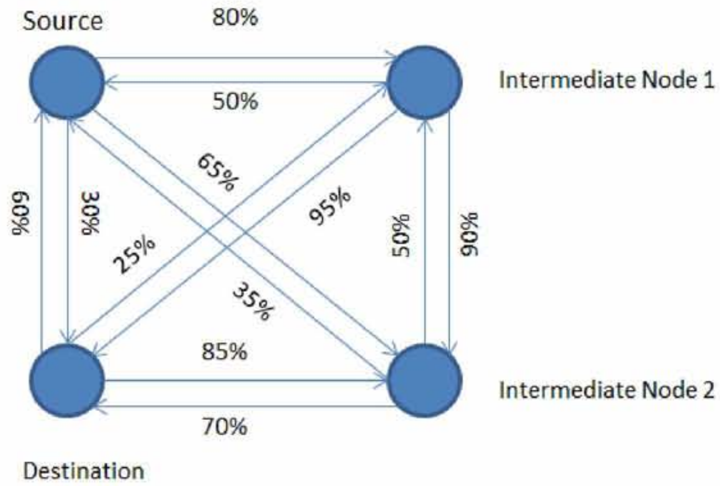


Figure 1. The starting network of 4 nodes with its links estimated,

The algorithm works as it follows:

After receiving all the packets required decoding the contents of the batch the destination transmits.

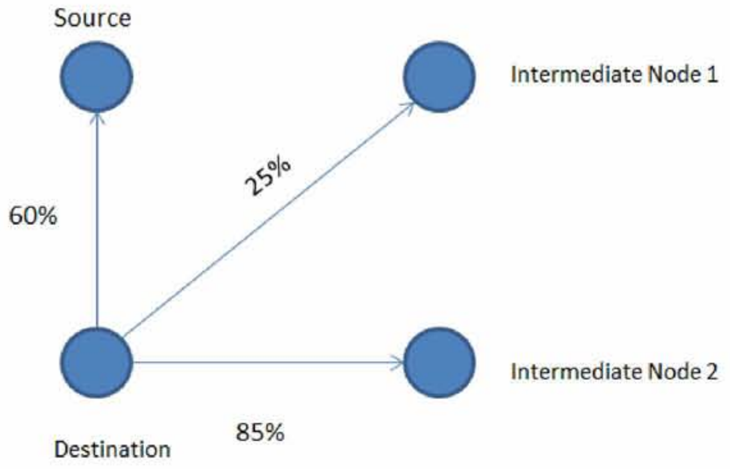


Figure 2. Destination sending probes at the beginning of the algorithm

When the other nodes hear that probe they "awaken" and send their respective probes when the 802.11 MAC permits.

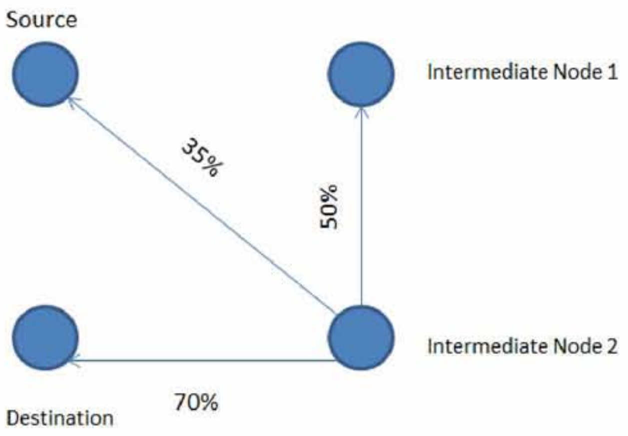


Figure 3. Node 2 transmits

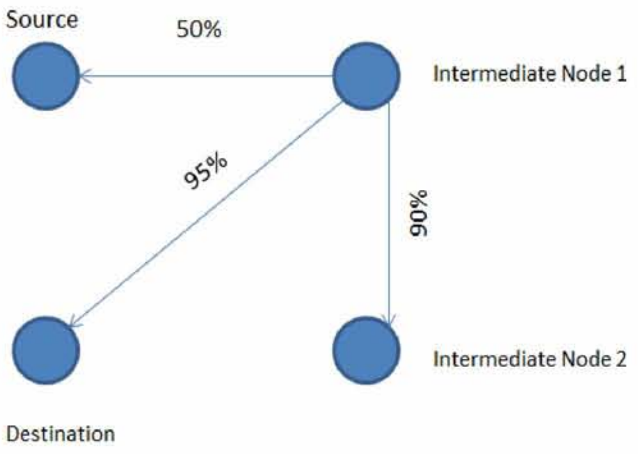


Figure 4 Node 1 transmits

In a fortunate situation all the probes will be received properly and the nodes will calculate the new link delivery ratio estimations

Let's now consider two scenarios of unfortunate events. In the first scenario some nodes fail to receive at most one probe packet. This means that every node will be able to hear successfully at least one probe packet in order for all nodes to "awake" and send their respective probes. Here's how the algorithm works in this situation:

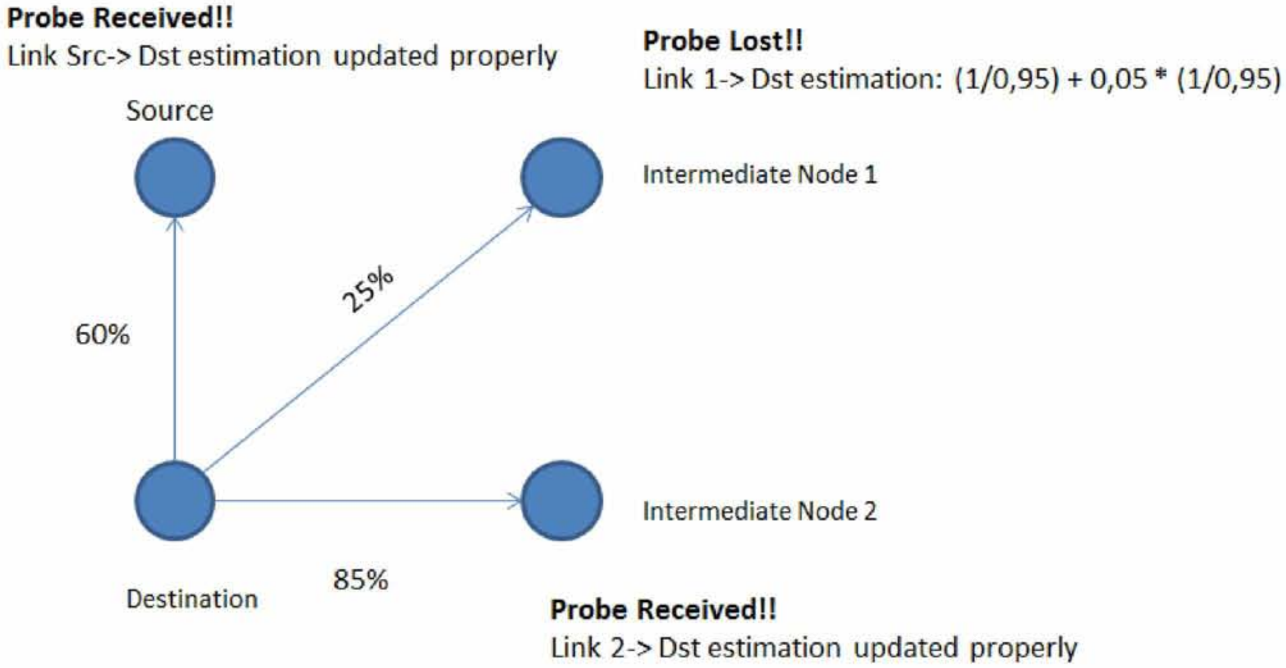


Figure 5. Step 1 of the algorithm

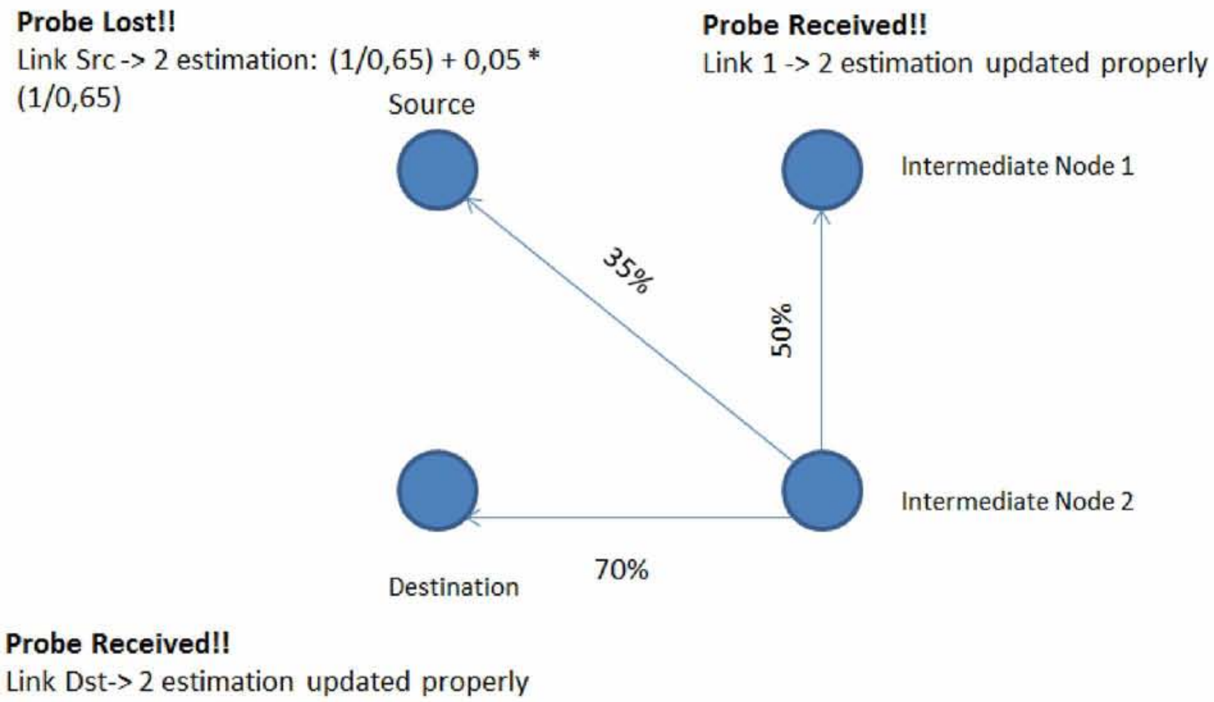


Figure 6.Step 2 of the algorithm

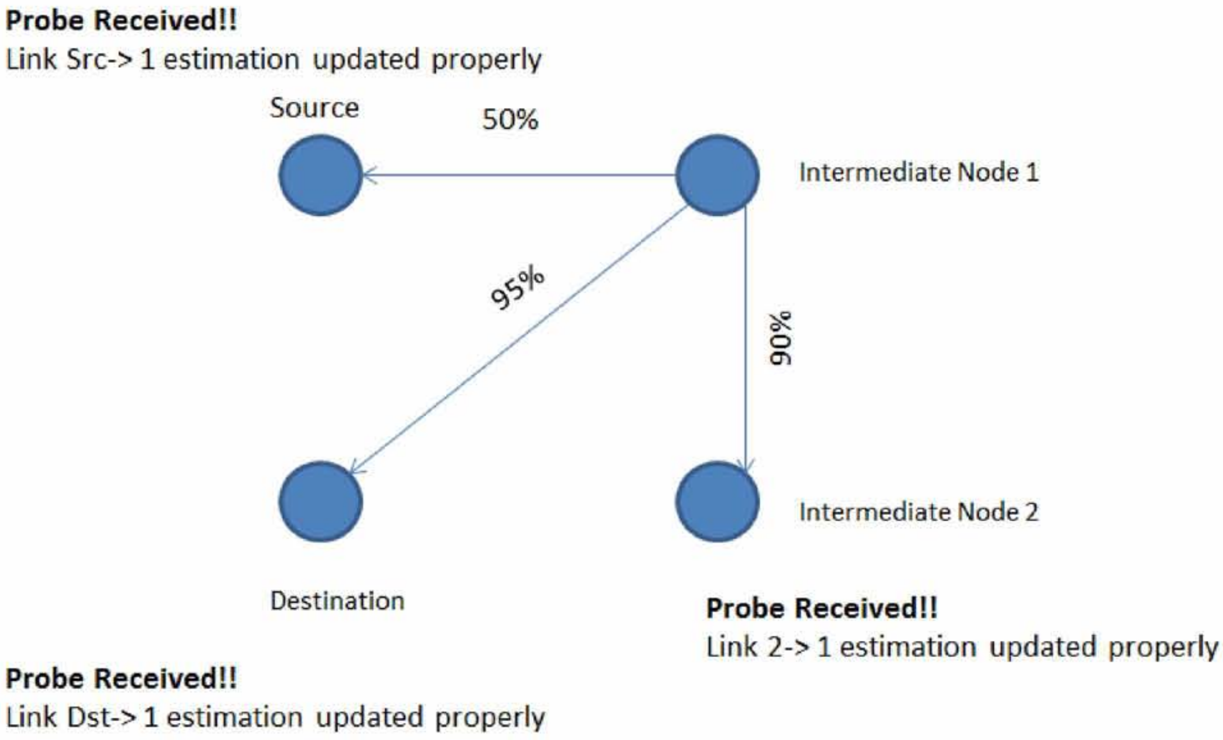


Figure 7.Step 3 of the algorithm

For the second scenario we assume that at least one node does not receive any probe packets when the first coded packet from another batch arrives. Our algorithm reacts to this as it follows.

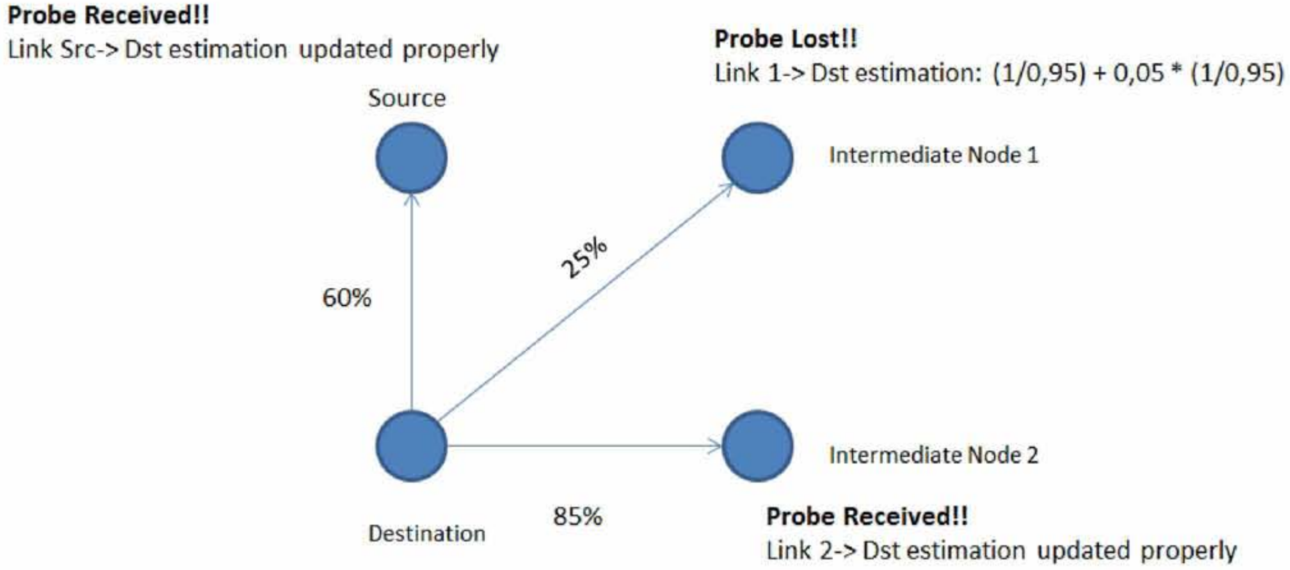


Figure 8.Step 1

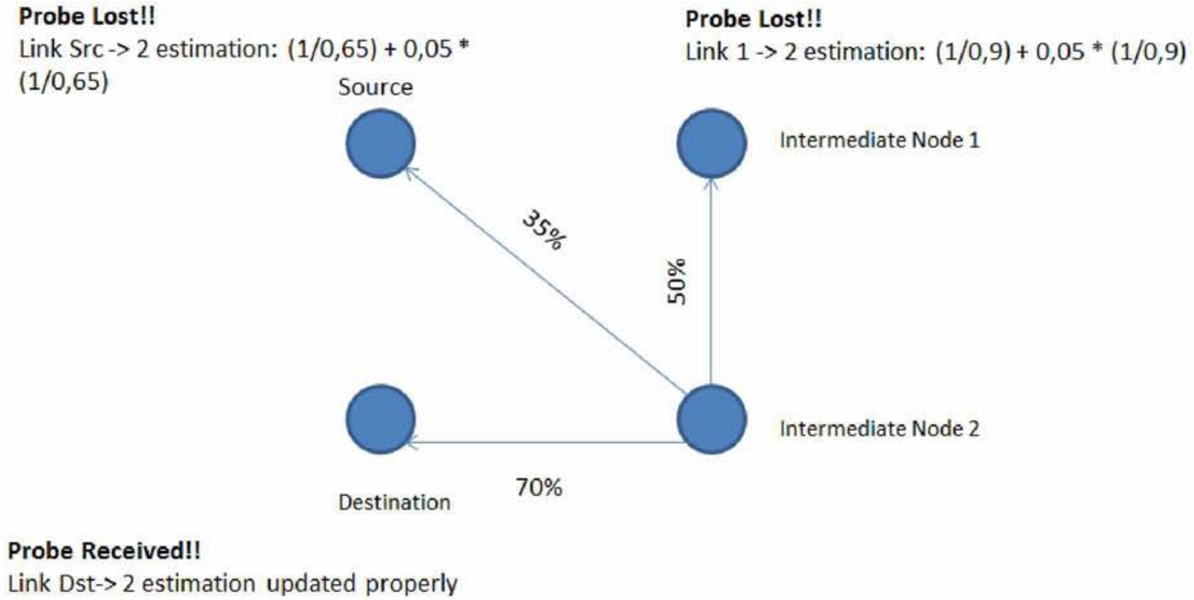


Figure 9.Step 2

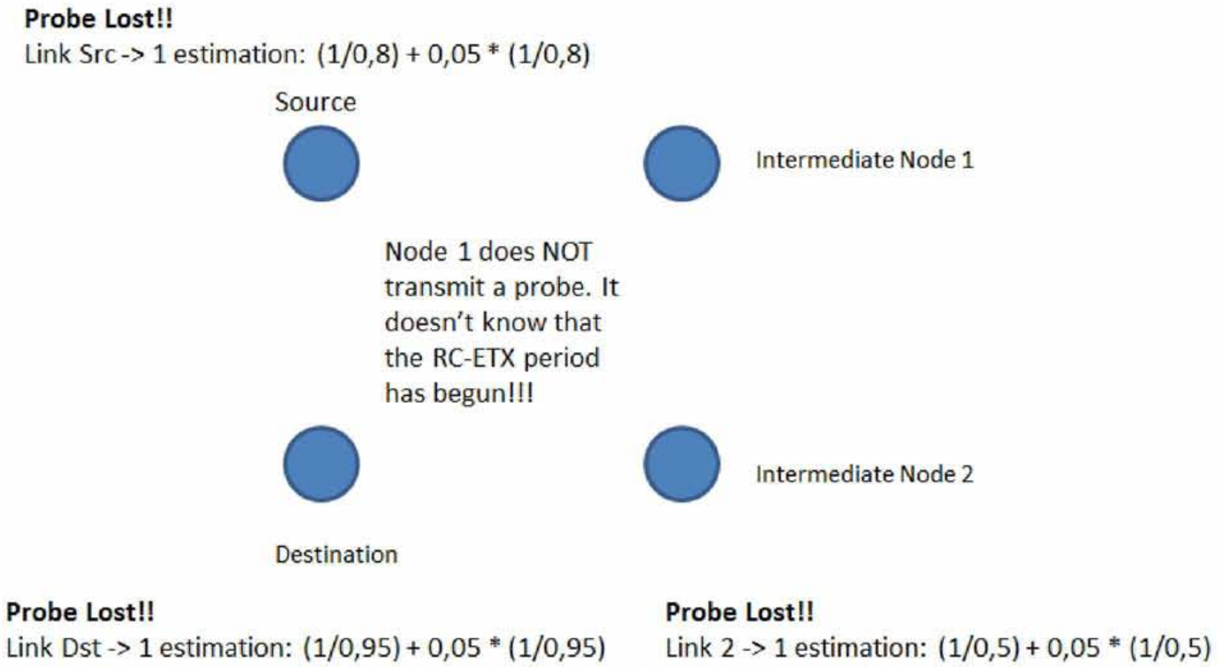


Figure 10.Step 3

4.3 EVALUATION OF THE ALGORITHM

Although at first look the above approach seems to be working there are a few issues that arise include the send-once-and-forget transmission of the probe packets. Another issue is that the above mentioned algorithm might not work well on non-static networks where links might have large variations during data transmissions that RC-ETX might fail to detect since it is applied at the end of each batch transfer. In addition to that we know that links aren't symmetric so a link with a high forward probability ratio might have a low reverse probability ratio so the probes sent to calculate the forward delivery ratio will be lost more frequently. We have to find some ways to improve our algorithm in order to somewhat alleviate these drawbacks.

As for the first problem we have introduced a heuristic solution that might not be optimal but it seems adequate since the simulation results show positive feedback. The solution is divided into three cases described below:

- If a node doesn't receive a probe packet but has a previous RC-ETX or ETX metric for a certain link:
  - It keeps the previous metric but increases it by 5%. This happens because probe packets are small and have increased chance to be received by a certain node. If a probe packet's reception fails it means either of two: the link between the two nodes has degraded or the probe was lost due to a rather unfortunate transmission. In each case we use the link's metric history to update the current link loss ratio. The above method penalizes more the lossy links who have a higher chance to degrade while

good quality links are less affected by that 5% degradation.

- If a node doesn't receive a probe packet and has no previous RC-ETX or ETX metric for a certain link:
    - It keep the metric value to 0 and takes no action
- If a node receives a probe packet and has no previous RC-ETX or ETX metric for a certain link:
    - It updates the link metric according to the newly received data using the RC-ETX calculation

For the second problem one proposed solution is to perform several ETX sessions pre-emptively, before data transmissions begin. We can find a heuristic algorithm that adjusts the degradation percentage which might also include negative values in order to capture the possibility of unfortunate packet losses who might happen randomly but don't necessarily mean that a link has degraded. This heuristic algorithm will take in consideration all of the pre-emptive ETX results and create a network link variation profile that will help the aforementioned adjustments.

Although the above suggested algorithm is not implemented in our current implementation of the RC-ETX algorithm it is proposed here as a future possibility.

Another proposed solution to the send-once problem is to attempt a normal unicast approach of the probe transmission. In that case many retransmissions might occur and introduce overall data transmission delays. Also the inclusion of ACKs and retransmissions increases the overall energy consumption of the RC-ETX algorithm making it less efficient than the heuristic approach mentioned before. This is supported by simulation results that show increased energy consumption than the heuristic approach which is still lower than that of the ETX but it might reach equally high consumption in certain scenarios.

4.4 UPDATED ALGORITHM EVALUATION

With the above improvements we manage to somewhat address the issues described in a previous subchapter. The solutions proposed though must not be taken as optimal since they are mostly heuristic approaches. Some issues continue to persist such as algorithm's inflexibility in highly variable links in contrast to ETX which is still more flexible.

In static networks though where links have small variances during packet transmissions RC-ETX shines with its estimation accuracy being almost identical to that of ETX and superior energy efficiency.

4.5 SIMULATION DESCRIPTION

In order to evaluate the conceived algorithm we have written a simulator in the c programming language which emulates a wireless network with a variable number of nodes and is capable of measuring end to end delay, consumed power and metric accuracy. Loss in wireless link is modeled by using Rayleigh channel fading. The SNR values required for the calculations are between 10 to 40 dB and their value is randomly selected for each link every time the simulation is run. It is important to note that the simulated network once initialized will keep the SNR values of its links unchanged based on the hypothesis that every link will have no fluctuations in terms of SNR and eventually stabilize after a period of time. BER derives from the link model and is calculated accordingly.

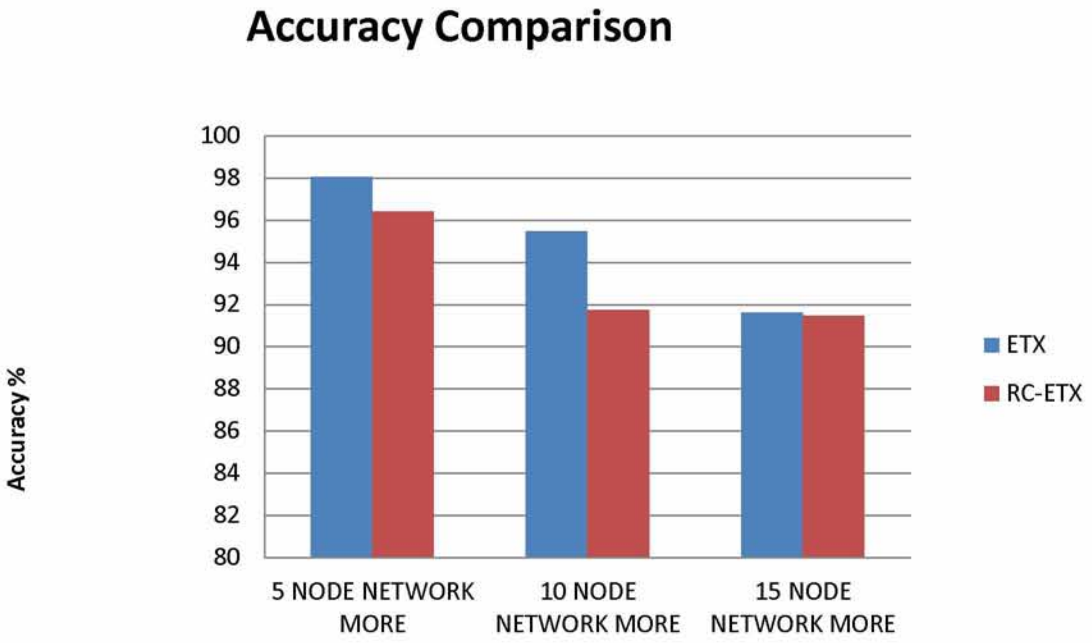
Data packets are of 256 bytes in size and are sent through the wireless medium by broadcasting at a 40Kbit/s rate. The total number of data to be sent from source is 25KB and is split in 100 packets. When MORE is initialized it splits the total number of packets to a total of 10 batches each of equal size. This means that we get 10 packets to be encoded each time a batch is being transmitted. MAC layer performs CSMA to allow each node to gain access to the medium and transmit its data.

For the ETX implementation we are using an ETX period of 1 second. This period occurs each time after 1 second of data transmissions. ETX probes are smaller than data packets but are sent through the medium with the same delivery chance as the data packets. The same goes for the RC-ETX probes.

4.6 TEST RESULTS

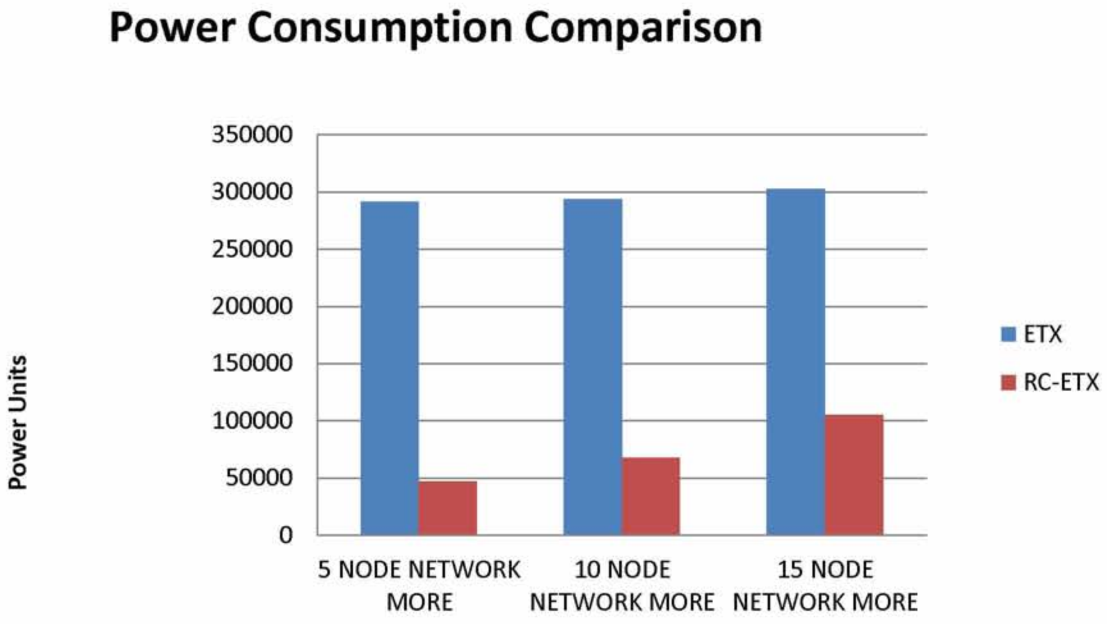
In our tests we compared the ability of each algorithm based on different network sizes all using MORE as their routing algorithm.

1) First we tested each algorithm's accuracy. By "accuracy" we mean how accurate is an algorithm at predicting an individual link's actual delivery probability ratio. It is the median value of the absolute difference between each link's actual loss ratio and the estimated one.



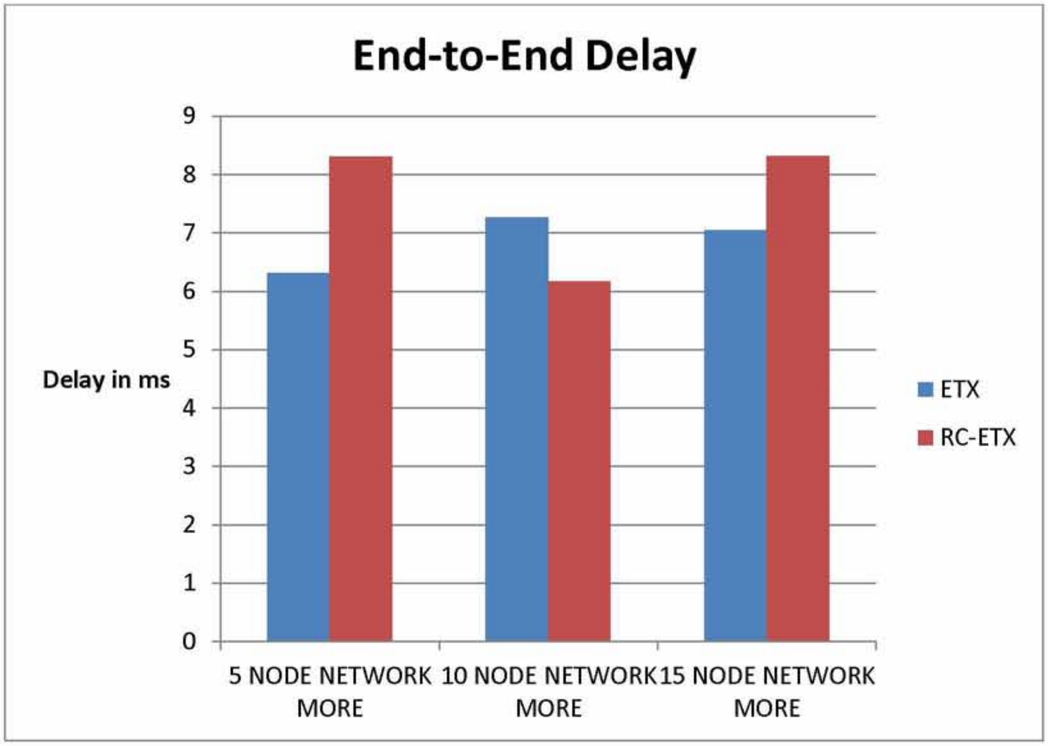
We notice that both algorithms perform almost the same under all the network sizes of our tests. This result is the expected one because both of these algorithms use almost the same type of data to calculate their respective metrics. We also notice that the RC-ETX can reach accuracy values of 100% in its expectations which is higher than that of the ETX although not very apparent since ETX falls short only a 1-2%. On the contrary in some scenarios it performs worse than ETX but the difference is still negligible.

2) The next test is about each algorithm's power performance. It measures how much power does each algorithm consume to calculate its metric. The power is measured in power units and not in watts but both results would be the same since power units are calculated by using a static power value multiplied by the total number of probe transmissions each method needs in order to calculate its link loss estimations.



Here is where our algorithm shines. While offering almost the same level of link loss ratio estimations it manages to outperform ETX in terms of power consumption. We notice a maximum difference of nearly 84%. One may argue that if we set the ETX periods to occur less frequently will reduce the overall consumption of the ETX algorithm. In the same manner we can increase the number of batches needed to perform an RC-ETX measurement instead of the per batch basis.

3) The final test is measuring the end-to-end delay. Here we want to see whether our new algorithm increases or decreases end-to-end delay compared to that of the ETX.



We observe that the end-to-end delay is almost identical to both and the small differences that appear on the graph are due to the randomness in each network's link loss probabilities and nothing more.

5. FUTURE WORK

As written above the RC-ETX is a very good algorithm to be used instead of ETX on wireless opportunistic networks. It's not perfect though and needs improvements, some of them have been already been discussed previously on this thesis.

- One thing we think it’s important is the need to test how this algorithm performs if we discard our send-and-forget method of sending probes and replace it with the normal unicast retransmissions including a retransmission cap.
- Another way of facing the above issue is by piggybacking information received from a probe to a node’s current probe packet
- We might need something more versatile. RC-ETX's versatility is based on the number and the size of the batches. This means that a high number of small batches like this in our test increase the metric's versatility since it takes a total of 11 measurements in our test while ETX with a frequency and a period of 1 second will take only 6-8 measurements depending on the total transmission time.
- Find a way to apply this algorithm to all types of networks.
    - Currently our approach is targeted for opportunistic networks that group data packets into batches and transmit them by broadcasting them through the wireless medium. While this works on this type of networks it doesn't work on the others since we don't take in account the reverse delivery ratio of a link. This ratio denotes the chance of successfully receiving an ACK after a packet's reception by its destination.

6. THANKS

First of all I would like to thank my supervisor on this thesis Mr. Antonios Argyriou lecturer at the university of Thessaly for his invaluable input and additional research on my thesis.

I'd also like to thank post-graduate student Dimitrios Kosmanos for his help on some issues concerning the Rayleigh channel simulation.

Finally I would like to thank everyone that encouraged me to complete this thesis and supported me in any way possible.

7. REFERENCES

[1] Trading Structure for Randomness in Wireless Opportunistic Routing
http://groups.csail.mit.edu/netmit/wordpress/wp-content/themes/netmit/papers/MORE.pdf

[2] A High-Throughput Path Metric for Multi-Hop Wireless Routing.
https://pdos.csail.mit.edu/papers/grid:mobicom03/paper.pdf

[3] Opportunistic Networks (ExOR)
http://en.wikipedia.org/wiki/ExOR_%28wireless_network_protocol%29

[4] Linear Network Coding and Random Linear Network Coding
http://en.wikipedia.org/wiki/Network_coding

[5] Linear Independence
http://en.wikipedia.org/wiki/Linear_independence

[6] Rayleigh Fading
http://en.wikipedia.org/wiki/Rayleigh_fading

[7] Rayleigh Modeling – BER for BPSK in Rayleigh channel
http://www.dsplog.com/2008/08/10/ber-bpsk-rayleigh-channel/

[8] Packet Broadcasting
http://en.wikipedia.org/wiki/Broadcasting_%28networking%29

[9] MAC Layer
http://en.wikipedia.org/wiki/Media_access_control

[10] Signal to Noise Ratio (SNR)
http://en.wikipedia.org/wiki/Signal-to-noise_ratio

[11] Matrix Inversion Methods
http://en.wikipedia.org/wiki/Matrix_inversion#Methods_of_matrix_inversion

[12] High Dynamic Destination-Sequenced Distance –Vector Routing for Mobile Computers
Charles E. Perkins, Pravin Bhagwat

[13] DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks
http://www.monarch.cs.rice.edu/monarch-papers/dsr-chapter00.pdf