An Application of deterministic Network Coding in MANETs

Leenta M.J. Grobler, Albert S.J. Helberg, Henri Marais, Coenraad C. Naudé
School for Electrical, Electronic and Computer Engineering
North-West University - Potchefstroom Campus
Tel: (018) 299-1961 Tel Fax: (018) 299-1977
leenta.grobler@nwu.ac.za

Abstract—Recent advances in methods to increase network capacity has lead to the introduction of a new concept called Network Coding. Network Coding holds the promise of increasing capacity in MANETs as well. However, very few practical results and implementations are available. In this paper we propose the use of deterministic Network Coding in a MANET. We present and describe the tools used to implement and investigate this new method.

Index Terms — Capacity, Mobile Ad-hoc Networks, Network Coding, Network Topology.

I. INTRODUCTION

THE universal need for better control over resources in communication networks is a problem that is studied continuously. Maximum network capacity needs to be defined and then utilized to ensure that as much information as possible is delivered in the most economic manner. One way in which this can be addressed, is by using Network Coding. Research on Network Coding to date led to a wide variety of theoretical results especially in wired networks. In wired networks, Network Coding can be implemented deterministically.

We also want to address the capacity and resource control problem in Mobile Ad-hoc Networks (MANETs). MANETs are complex networks of which the topology changes constantly and unpredictably. One documented implementation of Network Coding in MANETs is Random Network Coding [1].

In this paper, we investigate the opportunities that the properties of MANETs provide for practical implementation of deterministic Network Coding.

This paper is structured in the following manner:

We first give an introduction to MANETs. We then look at the maximum throughput capacity with the Min cut Max flow theorem.

Network Coding is explained together with the advantages possible by combining the inherent properties of MANETs and the advantages of Network Coding. We introduce our new method to implement deterministic Network Coding in MANETs. Finally we present a modified capacity calculation algorithm to test the efficiency of our new method.

II. MOBILE AD-HOC NETWORKS

A Mobile Ad-hoc Network (MANET) is a type of wireless network that typically consist of mobile routers and in some cases also laptop computers. These wireless mobile nodes are connected by wireless links to form a varying arbitrary network topology.

Because these nodes are free to move randomly and organize themselves arbitrarily, the topology may change rapidly and unpredictably. The management of ad-hoc networks is decentralized. That implies that each node present in the network act as a forwarding node, forwarding messages to other nodes. The selection of forwarding nodes changes dynamically with the topology. A MANET may operate as a standalone network, or be connected to a larger network such as the internet.

In other words a MANET is a network that is highly mobile, consisting of nodes with high processing power that receive frequent routing updates.

The advantages and disadvantages of MANETs can be summarized as follows [2]:

Advantages:

i. Adaptability
ii. Flexibility
iii. Efficient communication in environments with little or no infrastructure.

Disadvantages:

i. Vulnerable to attacks
ii. Congestion in the network and poor utilization of the network.

These disadvantages manifest as a result of a combination of factors: The use of an open medium, with a decentralized nature and a topology that changes dynamically, with poor physical security.

One of the biggest challenges in working with MANETs is to determine the network capacity. When the capacity is known, using [3, 4], we can make use of Network Coding to utilize this capacity,

In our research we focus on developing a new technique to reduce congestion in a MANET while improving the utilization using Network Coding.

III. MIN CUT MAX FLOW THEOREM

As an introduction to the maximum capacity of a network, we refer to the Min Cut, Max Flow theorem [5]. Alternative methods include Steiner tree packing, Linear Programming and a method called MSDA [3, 4].

Min-cut Max-flow theorem:

The maximum flow of information in a network is equal to the sum of the cut of the link capacities [5].

Using the theorem - we cut the network, separating the sender and receiver nodes (as can be seen in figure 1) to cross as little of the links as possible to determine the minimum cut. We can see that the minimum cut, and therefore the maximum flow (or maximum throughput capacity) of this network is equal to 2.

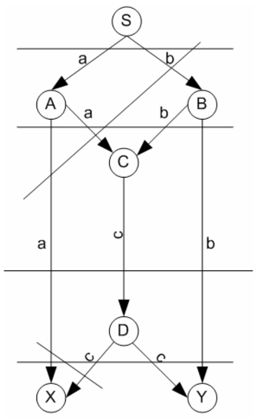

Fig. 1. Butterfly Network explaining Min Cut, Max Flow Theorem

The only way that we can utilize this maximum throughput capacity, is by using Network Coding [6].

IV. INTRODUCTION TO NETWORK CODING

A. *What is Network Coding?*

Network Coding is a field that was first introduced in 2000 [7] as a method to utilize the maximum capacity of a network and maximize the flow of information in that network. It suggested coding at packet level in wired P2P networks. The idea sprouts from research done in [8] on satellite communications using a source coding system which consists of multiple sources, encoders, and decoders.

Applications where Network Coding can be especially useful are MANETs, Power Line Communication as well as Wireless Sensor Networks.

B. *How does Network Coding work?*

We will now explain Deterministic Network Coding: We use the Butterfly network from [9], as depicted in figure 2, to explain the how Network Coding works. The links in the figure all have unit capacity and messages *a* and *b* are binary.

Two nodes, *A* and *B* need to transmit their messages to both Nodes *X* and *Y*. Each of the nodes can deliver their own message to the node that is connected to it, but have to route their messages through the network to reach the other node. When making use of traditional routing (Figure 2a), node *C* simply receives and replicate the information it receives from the previous sender node. In this case the two messages *a* and *b* will reach node *C* simultaneously. Node *C* will send out message *a* first, and then message *b*. Thus, at the end of a single arbitrary time unit, only node *Y* will have received both messages, while node *X* still has only message A. This results in a throughput of 1.5.

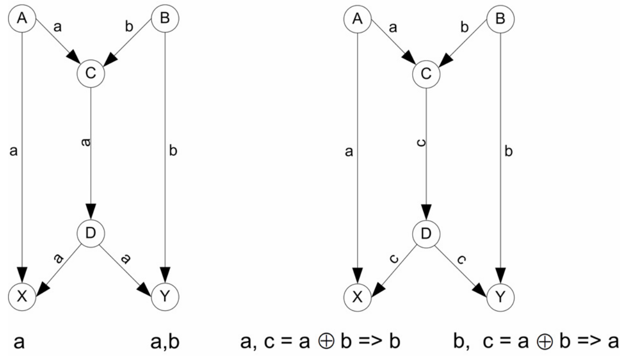

Fig. 2. Butterfly Network a) Without Network Coding b) With Network Coding

When we make use of Network Coding (Figure 2b), we give node *C* the capability to transmit a linear combination (logical XOR) of the binary messages *a* and *b*. Message *c* has the same length as message *a* and *b,* and is transmitted via node *D* to nodes *X* and *Y*. We then give nodes X and Y the capability to decode message *c* by using the other message it already received and solving the two linearly independent equations. In this special case, it merely means adding the single message that the node has already received to the network coded message. This time, by the end of a single arbitrary time unit, both nodes *X* and *Y* have both messages. Two messages were delivered, making the throughput 2. [7]

This method however changes the way node *C* works, because it has to form linear combinations of the messages it receives before forwarding it. It also requires nodes X and Y to have knowledge of the network topology and how the messages reaching it are encoded in order to deduce the two original messages from the messages it received.

An attempt to make Network Coding more practical for MANETs is Random Network Coding [1], but we want to see if we can use a known topology of the deterministic approach in a MANET. Because the deterministic approach to network coding requires little overhead, this approach may be useful to improve network utilization in a MANET.

C. *What benefits do we get from Network Coding*

The use of Network Coding in a network may provide the following benefits:

1) Throughput [7], [9], [11]:

The improved throughput in networks was the first major result of Network Coding.

If we refer to the throughput achieved with network coding in the deterministic example, we see that we have achieved the maximum throughput as calculated using the min-cut max-flow theorem.

2) Robustness [9], [11], [12]:

The robustness of the network refers to the ability of the network to remain functioning even though a link has failed completely.

3) Adaptability [11], [13]:

Adaptability is an important benefit when looking at MANETs, as this refers to the ability of the network to cope with nodes constantly joining and leaving the network, resulting in a constantly changing topology.

4) Security [11]:

The security benefit is an inherent benefit, seeing that linear combinations of data are sent over the network, and not the actual data. This benefit while useful, is however not sufficient. If a malicious entity listens long enough and receive sufficient messages to decode the information, the information can still be eavesdropped.

Thus we see that Network Coding can address many of the problems associated with MANETs.

V. CONCEPTUAL NEW IDEA

We found that there is currently no method that indicates where in a MANET or *how* exactly at that specific location deterministic Network Coding could be implemented. This led to our research question: *Can a known topology used in deterministic Network Coding be used to apply Network Coding Practically in MANETs?*

Our hypothesis is:

The unique mobility present in MANETs creates multiple opportunities for opportunistic implementation of deterministic Network Coding in the routing of information.

We propose to examine the connection matrix of a MANET, to see if we can find sub-matrixes that are known Network Coding networks. If we can find these sub-matrixes, we know where in the MANET the opportunity to implement Network Coding exists. Because we use known topologies we then also know how to implement Network Coding, i.e. which nodes should change in their functionality, and which should have information on the network topology. Our aim in this is to improve the local throughput of part of the MANET, by using deterministic Network Coding topologies in a part of the information path.

We foresee the following benefits when using this concept:

1. An improvement in the total throughput (better utilization of the network's capacity)
2. Lower occupation of the total network.
3. Improved quality of service, because of a lower delay in the network.

We propose the following method to illustrate this concept:

i. Select a Network Coding topology of which the gain and capacity is known.
ii. Derive the connection matrix of this Network Coding topology.
iii. Determine the connection matrix of the larger MANET from a suitable distance vector routing algorithm;
iv. Search the larger MANET matrix for the known topology matrix.
v. Implement Network Coding at the appropriate nodes.
vi. Re-iterate steps (iv) and (v) after a routing update.

In order to evaluate whether or not it is worth it to implement deterministic Network Coding in the MANET, we compare the capacity of the MANET before and after the opportunistic implementation of Network Coding.

VI. IMPLEMENTATION OF METHOD

To demonstrate that implementation of this concept is practically possible, we will now use the Butterfly topology as our known topology, and search for it in a random MANET. We do this as follows:

A. Initial steps (Steps i – iii)

We translate the Butterfly network in Figure 2 to a 6x6 connection matrix:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

We then create a connection matrix of a random MANET of 7x7 with a connectivity of at least 3 connections per node, representing the state of the network for one stream of information. We use a 7x7 network, because it is currently the size of the largest physical testing facility in South-Africa (at the MERAKA institute), but we also test our technique in larger networks, to test scalability of the technique. This is sufficient for this implementation, because the size of the random MANET is one order bigger than the matrix that we search for. We scan the large matrix, using the method described in *step iv,* for all permutations of the smaller 6x6 matrix.

Once we have found the smaller matrix-location in the larger matrix, we know (referring to figure 2b) that node C should have encoding capability, thus the node should logically XOR the information received from nodes A and B. Nodes X and Y should have decoding capability, that is, they should derive the two original messages from the information received.

B. Finding the required topology (Step iv)

We generate a random *n x n* connection matrix with a connectivity of at least 3. We then want to locate the butterfly topology inherently present in the random matrix. In order to locate the butterfly matrix within the larger network connection matrix several methods can be used. The simplest of these methods would be iterative looping. The fundamental concept of iterative looping is a nested looping structure. This effectively moves a window across the network connection matrix and continually compares the data in the current window to that of the desired data, in this case the butterfly matrix.

Although this method is accurate it is not desired since it scales poorly, and is computationally intensive. For the reasons stated above we propose using a different method.

Our method is based on the two dimensional cross correlation of the network connection matrix and the butterfly matrix. The cross correlation of two datasets can be seen as the similarity of those two datasets, and is commonly used to search for patterns in a random dataset by forming the cross-correlation to known patterns. This technique finds application in digital signal processing (DSP) as well as image processing.

The concept of cross-correlation is extended to two dimensions with the network connection matrix being used as the random dataset and the butterfly matrix being used as the known pattern. The result of the cross-correlation operation *xcor(connection, butterfly)* results in a new matrix M with values indicating the level of cross-correlation. In order to extract the correct values, the auto-correlation of the butterfly matrix is formed. By scanning M for this value the location of the butterfly matrix can be determined.

Advantages of using the cross-correlation method is scalability, as the method remains computationally efficient when using larger matrices. A further advantage is the ability to detect where partial Butterfly structures occur within the network connection matrix. These partial structures then have the potential to become butterflies if the topology of the network changes slightly.

Both methods were implemented in Matlab in order to compare both the accuracy as well as the computational efficiency. For small matrices of less than 40x40 both methods perform equally well in terms of computation time. For larger matrices of 100x100 the cross-correlation method's computation time is significantly less than that of the iterative looping. As the matrix dimensions increase the computational advantages of the cross-correlation method increases.

We can then implement Network Coding at the appropriate nodes. The search is reiterated after possible topology changes. *(Steps v and vi).*

VII. CAPACITY CALCULATIONS

We define capacity of a MANET as: *Capacity is the number of 1-hop sessions that share the same channel. [3, 4]*

In order to test the improvement that our idea offers, we use a technique called MSDA [3, 4]:

MSDA uses the original generated connection matrix as the 1-hop adjacency matrix $A^1_{nxn} = A_{nxn}$. We then define a delay of k and use exact multiplication of A_{nxn} to calculate A^k_{nxn}.

It is then required to calculate all the *k-hop* paths. We do this with the following new algorithm:

1. We look in the upper-triangle of A^k_{nxn} for a value equal to k and store the column numbers of the start and end nodes (as the first and last values) in a vector *Path* of size *k+1* as the source and sink nodes.
2. We move to the row number corresponding to the column where the sink was found and search for the value *k-1*. Ignore the columns of the source and sink as well as the values on the diagonal of the A^k_{nxn} matrix. This value is stored as the second value in the vector *Path.*
3. We repeat the search in the corresponding row of this node's column for the value *k-2*, to fill the second to last element in *Path.*
4. We repeat this process until vector *Path* is filled and then perform a check that the last found node is indeed connected to the first node in *Path.* If it is not connected, the path is deleted.
5. All the generated *Path* vectors are then stored in *PathSet.*

Continuing with MSDA, for every node in *PathSet*, the probability that the node is present in *PathSet* is then calculated, and the sum of the probabilities on a path are then used to calculate the probability of each path.

The path with the lowest probability is then selected and all the paths containing any of the nodes that are present in the selected path are then deleted.

By doing this, we find all the independent *k-hop* paths in A_{nxn}.

This result is used in the MSDA algorithm, after which, we can calculate the capacity of the MANET as in [3]:

$$N_s = \min\left[\left\lfloor \frac{BW_{node}}{BW_{packet}} \right\rfloor, \frac{D_E(n^2 - n - n_0)}{\sum_{i \in A^k_{nxn}} i - n}\right] \quad (1)$$

Where $\lfloor BW_{node} / BW_{packet} \rfloor$ is the number of one-hop sessions that the channel can support, D_E is the end-to end delay, n is the number of nodes and n_0 is the number of 0's in the connection matrix A_{nxn}.

To calculate the benefit of using the Butterfly Network Coding network we then use the location of the Butterfly matrix in the larger matrix, and modify the MSDA algorithm in the following manner:

We ensure that the paths with node combination corresponding to nodes C and D in figure 2 are arranged at the top of *PathSet* to increase the probability that they are selected as a path. We also modify the algorithm not to delete *k-hop* paths with that specific node combination..

With this modification included, we again calculate the capacity, using (1).

Finally we compare the calculated capacities of the MANET to see the improvement of implementing Network Coding in this way.

VIII. PRACTICAL RESULTS

We implemented our method as described in section IV in networks with 6, 7, 8, 9, 10 and 11 nodes respectively.

We generated 100 random connection matrices for each size network. Our random matrix generator generated 100 unique matrices, checking that a particular connection matrix is not used more than once, but not necessarily that it is not a permutation of a matrix that was already tested.

The generated matrices also had connectivity (*C*) of exactly 3, meaning that each node was connected to 3 other nodes. The connectivity of three is the minimum connectivity that will allow the possibility of a butterfly network to exist (refer to figure 2), but does not guarantee the existence of a butterfly network in large networks. Results will be better for networks with higher connectivity.

The results of this implementation can be seen in table 1.

TABLE I
IMPLEMENTATION RESULTS

Nodes:	C:	Networks Searched:	Networks with Butterflies	Total Butterflies
6	3	100	100	1234
7	3	100	100	1608
8	3	100	100	2140
9	3	100	100	893
10	3	100	100	2038
11	3	100	100	1252

In all of the networks butterflies were found (being small networks). In most instances far more than one butterfly network was found, although these are not necessarily disjoint or unique butterfly networks.

IX. DISCUSSION OF RESULTS

We find the following four improvements in the network:

i. More available paths:
For each butterfly network present, we have the possibility of one extra path available, sometimes more provided that two butterfly networks do not make use of the same link.

ii. Better utilization of the network
This can be seen as a qualitative improvement.

iii. We can define an upper limit for the network throughput.
The communication flow that passes through the paths making use of Network Coding has an upper limit of the throughput of the network code. In the best case, the paths used in the flow only meet in the Network Coding section and thereafter disjoin, thus the Min Cut of the flow is situated in the Network Coding section. In the worst case, the Min cut occurs outside the Network Coding section and has a lower value.

iv. This method is energy efficient, because fewer transmissions are needed to deliver the same amount of information.

This method proves to be successful, provided that we use a routing protocol that knows what the whole network looks like – such a protocol for MANETs is currently being developed.

The cost of using this method is the following:

i. Clever nodes:
ii. Route establishment delay
iii. A high connectivity is required

These drawbacks are compensated for when the MANET is implemented using Laptops in a close vicinity to each other. Laptops inherently have the high processing power to perform the computations for these "clever nodes" and are able to compensate for the route establishment delay.

X. CONCLUSION

The changing nature of MANETs offers many possibilities for the opportunistic implementation of Network Coding. It is impossible to guarantee a specific increase in capacity due to the constantly varying topology of the MANET. Theoretically this increase in topology can be as high as the maximum throughput capacity, but in practice this is seldom seen.

We have presented in this paper the necessary tools for the network planner to determine whether our Network Coding method can be useful or not for a specific implementation.

REFERENCES

[1] T. Ho, R. Koetter, M. Médard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," In *Proceedings of the IEEE International Symposium on Information Theory* June 2003, page 442, Yokohama, Japan, 2003.

[2] A. Mitrokotsa, N. Komninos, C. Douligeris, (2007), "Intrusion Detection with Neural Networks and Watermarking Techniques for MANET", In *Proceedings of IEEE International Conference on Pervasive Services 2007* (ICPS'07), July 15-20, 2007, pp. 118 - 127 Istanbul, Turkey, 2007.

[3] J. Zhang and W. K. G. Seah, "Topology–based capacity analysis for ad hoc networks with end-to-end delay constraints," In *Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication,* 31 May-2 June 2004, vol.2, pp. 541- 544, 2004.

[4] Ning Li, Yan Guo, Shaoren Zheng, Chang Tian, Jun Zheng, "A Matrix-Based Fast Calculation Algorithm for Estimating Network Capacity of MANETs," La Jolla, CA, pp. 407-412, *Systems Communications* (ICW'05, ICHSN'05, ICMCS'05, SENET'05), 2005.

[5] P. Elias, A. Feinstein, and C. E. Shannon, "Note on maximum flow through a network," *IRE Trans. Information Theory*, vol. 2, pp. 117–119, 1956.

[6] S. A. Aly, V. Kapoor, J. Meng and A. Klappenecker, "Bounds on the Network Coding Capacity for Wireless Rondom Networks", *Information Theory and Applications Workshop*, Jan. 29 2007-Feb. 2 2007, pp. 231-236 ,2007.

[7] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow", *IEEE Transactions on*

Information Theory, vol. 46, July 2000, pp. 1204-1216, 2000.
[8] R. W. Yeung "Distributed Source Coding for Satellite Communications", *IEEE Transactions on Information Theory*, May 1999, vol. 43, pp. 1111-1120, 1999.
[9] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding", IEEE Transactions on Information Theory", February 2003, vol. 49, pp. 371- 381, 2003.
[10] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright and K. Ramchandran "The Benefits of Network Coding for Peer-to-Peer Storage Systems", *NetCod Workshop*, January 2007.
[11] C. Fragouli, J. Widmer, J. Le Boudec "Network Coding: An instant primer", *ACM SIGCOMM Computer Communication Review*, January 2006, vol.36, pp. 63 - 68 , 2006.
[12] R. Koetter and M. Médard, "Beyond routing: an algebraic approach to network coding", In *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol.1 , pp. 122- 130, 2002.
[13] C. Fragouli, J. Widmer, J. Le Boudec "On the Benefits of Network Coding for Wireless Applications", *Netcod*, 2006.

Leenta M. J. Grobler completed her Bachelor of Engineering degree in 2006, in Computer and Electronic engineering at the Northwest University – Potchefstroom Campus.

She is currently a second year Masters Degree student in Computer Engineering for TELKOM COE at the Northwest University, Potchefstroom campus.

Ms. Grobler is registered with ECSA as a candidate engineer and a member of the SAIEE.