TOWARDS MINIMISING THE COEFFICIENT VECTOR OVERHEAD IN RANDOM LINEAR NETWORK CODING

Chamitha de Alwis, H. Kodikara Arachchi, Anil Fernando and Ahmet Kondoz

I-Lab, CVSSP, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom
Email: {C.Dealwis, H.Kodikaraarachchi, W.Fernando, A.Kondoz}@surrey.ac.uk

ABSTRACT

Network Coding is a promising approach to increase network throughput and robustness to facilitate high volume traffic. Performing network coding in dynamic network structures requires transmitting coding coefficients for information sinks to decode network coded packets. Compared to the packet sizes used in practical networks, the size of coefficient vectors can be significant. This paper exploits the properties of small and medium sized networks and proposes a novel approach to minimise the coefficient vector size of network coded packets. Simulation results exhibit better compression of coefficient vectors over existing algorithms for small and medium sized networks.

***Index Terms*—** Random Linear Network Coding, Coefficient Compression

1. INTRODUCTION

Due to increased throughput and enhanced robustness offered by network coding, it has become one of the prominent areas in networking research. Network coding extends information coding to intermediate nodes [1]. Information packets are treated as streams of symbols from a certain finite field. Network nodes combine packets over the considered finite field and send out the combination. Information sinks are required to possess the knowledge on how network coding is performed on the packets they receive, in order to decode the original information. This is not feasible in practical networks due to node failures, link failures, packet losses, packet jitter and network topology changes. Random linear network coding extends network coding to be used under dynamic network structures[2], [3]. This is performed by appending the coefficient vectors in front of the coded packets to provide information on how packets are coded. These coefficients are updated at each node where network coding is performed. Practical network coding splits the number of packets to be coded using random linear network coding in to fixed sized groups called generations [4]. The number of packets belonging to a generation is called the generation size. However, the approach of sending coefficient vectors comes at the cost of additional bandwidth.

Consider a network coding scheme which uses a generation size of 100 and a Galois field 2^8. This scheme requires coding vectors to accommodate 100 bytes per packet. Even considering a larger packet size of 1400 bytes, over 7% of payload data is required to transmit coefficient vectors. Furthermore, doubling the generation size or using a larger finite field of Galois field 2^{16} results in doubling the coefficient vector size.

As a solution, a method named Subspace coding is introduced in [5], which, does not require transmitting coefficient vectors. Subspace coding uses a particular subspace selected by the information source to convey information. Sinks can decode the coded information by deciding the particular subspace. However, for larger packet lengths, the information rate achieved by subspace coding becomes the same as sending coefficient vectors [6]. Furthermore, designing subspace codes also becomes challenging for a multi-source networks [7]. Another approach to compress network coding vectors is proposed in [7]. The authors argue that the packets received at information sinks may only contain linear combinations of several source packets which are only a fraction of the number of source packets. Therefore, the number of coefficient vectors transmitted is limited by limiting the number of packets that are allowed to be combined in a network. However, depending on the network structure and the generation size, the difference between the number of source packets included in a linear combination received by a sink and the number of source packets, may be narrow. This will significantly reduce the gain achieved by the compressed coding vectors algorithm proposed in [7]. In addition, limiting the number of packets being combined may decrease the probability of a redundant combination being useful for multiple sinks in a network with packet losses.

In response, this paper presents a novel approach to minimise the size of coefficient vectors in a network coded packet for small to medium sized networks such as Abiline [8], JANET [9] and UUNET UK [10]. This type of networks is used as backbone networks, connecting a number of core network nodes, to carry high volume traffic. In the proposed approach, network coding is performed in the information source using a set of coefficient vectors which are also known to information sinks. The matrix containing the coefficient vectors is identified by the term coefficient matrix. Instead of sending the coefficient vectors itself, coefficient information or information indicating which row/rows of the coefficient matrix are used to generate the particular linear combination, is sent along with the coded packet. Coefficient information is further compressed using a lossless compression scheme before transmitting. Intermediate nodes decompress the compressed coefficient information, combine received packet combinations, update and compress coefficient information and send the new packet combination along with updated coefficient information to the next hops.

The rest of the paper is organised as follows. Section 2 elaborates the novel approach presented in this paper. Section 3 presents and discusses the simulation results comparing the proposed method to state-of-the-art. Section 4 concludes the paper.

2. METHODOLOGY

2.1. Information Source

The source selects n number of packets $(x_1, x_2, \dots x_n)$, where, each packet is a $1 \times p$ vector of elements from a Galois field 2^F, where, F is the field size. These n number of packets belong to the generation g. Using an $m \times n$ coefficient matrix $(m \geq n)$, where, each coefficient is also a non-zero element from the same Galois field 2^F, the source combines n packets to generate m linear combinations $(y_1, y_2, \dots y_n)$.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}_{m\times 1} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,1} & \cdots & c_{2,n} \\ \vdots & \vdots & & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m.n} \end{bmatrix}_{m\times n} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1} \quad (1)$$

The coefficient matrix is the $m \times n$ portion of a larger coefficient matrix with elements of the Galois field 2^F. This matrix is identified as the super coefficient matrix and is known to all network sources and sinks. The super coefficient matrix is generated in a way such that the rank of the $n \times n$ portion is n. This offers the flexibility to change m, n based on the network condition. $m - n$ number of redundant combinations will be transmitted by the source per each generation. The super coefficient matrix consists of non-zero elements to increase the probability of a redundant combination being useful for multiple sinks in a network with packet losses.

In conventional random linear network coding, the coefficients that are used to generate the particular combination will also be sent along with the packet. For instance, the combination y_1 is transmitted as,

$$[c_{1,1} \quad c_{1,2} \quad \dots \quad c_{1,n} \quad | \quad y_1]$$

Since coefficients are selected from a Galois field 2^F, each coefficient will accommodate F number of bits. However, in the proposed approach, instead of appending the coefficient vectors, k number of bits per coefficient are used to indicate which row/rows of the coefficient matrix are used to generate the particular combination *i.e.,* coefficient information. Thus, the combination y_n with coefficient information is represented in the form of,

$$[\mathrm{i}_{n,1} \quad \mathrm{i}_{n,2} \quad \cdots \quad \mathrm{i}_{n,m} \quad | \quad \mathrm{y}_n]$$

For small to medium sized networks, similar to the ones specifically concentrated in this paper, a value of $k < F$ number of bits per row of the coefficient matrix will be sufficient to represent coefficient information without affecting the decoding rate. This will result in a ${k.m}/{F.n}$ times compression of the coefficient vectors. Furthermore, consider the argument stated in [7], which states if packets received at information sinks may only contain linear combinations of several source packets which are only a fraction of the number of source packets. This argument is well suited for small and medium sized networks considered in this paper. In this case, it is understood that majority of the elements in the coefficient information will be zero. In such cases, coefficient information can be compressed using a lossless compression algorithm. A simple compression algorithm that can be used is to indicate the position of non-zero elements and their values. The position of a non-zero element can be indicated using $\left\lceil {\log(generation\ size)}/{\log(2)} \right\rceil$ number of bits and the value (representing $value - 1$ is sufficient since only non-zero elements are considered) can be represented by k number of bits. However, depending on the coefficient information, a more suitable lossless compression algorithm can be selected out of several suitable lossless algorithms, such as Huffman coding, Golomb coding, arithmetic coding, etc. if necessary. If compression requires sending more bits than it would require to send the uncompressed coefficient information, the smaller of either the compressed or uncompressed coefficient information can be sent.

First a bits of the stream are used to indicate the compression algorithm used. The value of a depends on the number of compression algorithms considered by network nodes. These a bits are identified as the compression method identification flag. If no compression is performed, the compression method identification flag is set to null.

2.2. Intermediate Nodes

When an intermediate node receives a single combination from a generation, it will simply forward it to the next hops. If a node receives multiple combinations from the same generation, it adds the respective elements of the linear combinations over the considered finite field. If an intermediate node receives l combinations $(y'_1, y'_2, \dots y'_l)$, the output combination y'' is generated as,

$$y'' = [1 \quad 1 \quad \cdots \quad 1]_{1\times l} \begin{bmatrix} y'_1 \\ y'_2 \\ \vdots \\ y'_l \end{bmatrix}_{l\times 1} \quad (2)$$

Using the compression method identification flag, the particular algorithm used to compress the coefficient information can be identified and the compressed coefficient information is decompressed using the appropriate decompression algorithm. Then the coefficient information ($i'_1, i'_2, \dots i'_l$) is updated by separately adding the m number of k bits belonging to each combination together. The updated coefficients are,

$$[i'_1 \quad i'_2 \quad \cdots \quad i'_m]_{1\times m} = [1 \quad 1 \quad \cdots \quad 1]_{1\times l} \begin{bmatrix} i'_{1,1} & i'_{1,2} & \cdots & i'_{1,m} \\ i'_{2,1} & i'_{2,2} & \cdots & i'_{2,m} \\ \vdots & \vdots & & \vdots \\ i'_{l,1} & i'_{l,2} & \cdots & i'_{l,m} \end{bmatrix}_{l\times m} \quad (3)$$

Random linear network coding weighs received combinations using randomly generated coefficients from the Galois field 2^F before combining. This can increase the probability of the output packet being useful (increase the rank of received coefficient vectors) for sinks. However, as the authors concentrate on less complicated small and medium sized networks, addition without weighing is performed. Albeit, it is still possible to weigh combinations using coefficient values less than k. The output combination with coefficient information will be in the form of,

$$[i''_1 \quad i''_2 \quad \cdots \quad i''_m \quad | \quad y''_1]$$

The node selects the best algorithm to compress the coefficient information and compresses coefficient information. The compression method identification flag is updated accordingly. Then the compressed coefficient information along with the linear packet combination is transmitted to the next hops.

2.3. Information Sinks

An information sink first decompresses the compressed coefficient information. Using coefficient information, the sink can identify which of the coefficient vectors in the coefficient matrix have been used in generating each combination. The coefficient matrix is readily available in the sink. If the decompressed coefficient information of a packet received by an information sink is,

$$[\hat{i}_1 \quad \hat{i}_2 \quad \cdots \quad \hat{i}_m]$$

the coefficient vector of the received packet is,

$$[\hat{c}_1 \quad \hat{c}_2 \quad \cdots \quad \hat{c}_n]_{1\times n} = [\hat{i}_1 \quad \hat{i}_2 \quad \cdots \quad \hat{i}_m]_{1\times m} \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,1} & \cdots & c_{2,n} \\ \vdots & \vdots & & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m.n} \end{bmatrix}_{m\times n} \quad (4)$$

The sink collects n packet combinations ($\hat{y}_1, \hat{y}_2, \dots \hat{y}_n$), with linearly independent coefficient vectors. Then the original packets are calculated as,

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \end{bmatrix}_{n\times 1} = \begin{bmatrix} \hat{c}_{1,1} & \hat{c}_{1,2} & \cdots & \hat{c}_{1,n} \\ \hat{c}_{2,1} & \hat{c}_{2,1} & \cdots & \hat{c}_{2,n} \\ \vdots & \vdots & & \vdots \\ \hat{c}_{n,1} & \hat{c}_{n,2} & \cdots & \hat{c}_{n.n} \end{bmatrix}_{n\times n}^{-1} \cdot \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}_{n\times 1} \quad (5)$$

If no packet errors occur, the calculated information will be equal to the original packets.

3. RESULTS AND DISCUSSION

To analyse the gain achieved by the proposed algorithm in compressing coefficient vectors, simulations are carried out in a network topology similar to that of the Abiline network. Fig. 1. illustrates the simulated network topology. In the simulated network, the information source, node 0, multicasts information to information sinks located at node 5, node 7 and node 9. Simulations are performed for different generation sizes at Galois fields 2^8 and 2^{16}. Based on the network topology, a k value of 2 bits per coefficient is selected empirically to best suit the considered simulation scenario. Coefficient weighing is not performed. The compression algorithm explained in Section 2 is considered. Since only one compression algorithm is considered, $a = 1$ bit is sufficient to represent the compression method identification flag. $m - n = 2$ redundant combinations are transmitted per each generation. Results are compared against classical random linear network coding and the compressed coefficient vector approach proposed in [7].

For the convenience of analysis, header size is assumed to be similar to the number of bytes required to transmit either coefficient vectors or coefficient information. Fig. 2.a. and Fig. 2.b. illustrates the header sizes in bytes of the proposed algorithm, the compressed coding vector approach and classical random linear network coding, for Galois fields 2^8 and 2^{16}, respectively. It should be noted that simulations prove that the proposed algorithm does not affect the decoding rate of network coded packets compared to classical random linear network coding. This lays a fair ground to compare the required header size of the proposed algorithm with competing algorithms.

Analysing Fig. 2.a. and Fig. 2.b., it is observed that the header size in classical random linear network coding increases linearly with the generation size. The upper and lower bounds of the header size of the proposed algorithm also increases with the generation size in steps. This occurs due to the requirement of additional bits to indicate the position of non-zero elements while compressing coefficient information. However, the rate at which the upper and lower bounds of the proposed algorithm increase is significantly low compared to both random linear network coding and the compressed coefficient vectors approaches. Therefore, the header size of the proposed algorithm remains significantly

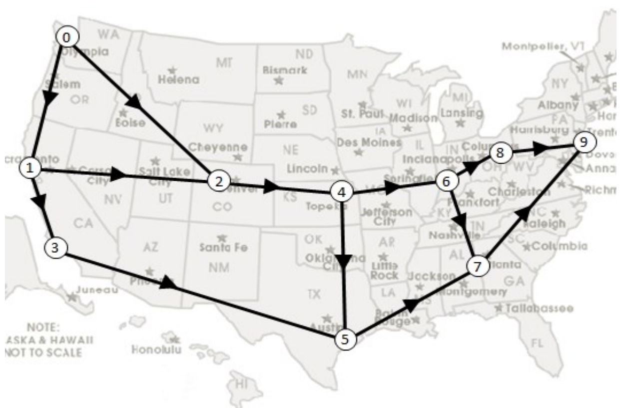

Fig. 1. Abilene Network Topology used to simulate the proposed algorithm.

smaller compared to other two approaches. In addition, it can also be observed that the header size of the proposed algorithm is not dependent on the size of the Galois field used, unlike the other two algorithms. Unlike the compressed coefficient vectors method, which limit the number of packets that are combined at the information source, packets sent from the source using the proposed algorithm are linear combinations of all the m packets in the generation. This is performed by using non-zero elements from the Galois field 2^F in the super coefficient matrix. This increases the probability of a redundant combination being useful for multiple sinks in a network with packet losses. Furthermore, as compression is performed on the coefficient information, the compression gain depends on how packets are combined in the network. Since authors concentrate on small to medium sized networks where the number of source packets that get combined are limited, a higher compression gain can be achieved. $k < F$ number of bits are sufficient to represent elements in the coefficient information for this type of networks. The value k should be carefully selected ensuring that a combination received by an information sink does not have the same source packet combined at intermediate nodes more than 2^k times.

In the proposed method, network nodes are required to perform compression and decompression operations. Therefore, compression algorithms should be selected considering the computational capabilities of network nodes. However, as networks evolve to software defined networks, the usage of complex compression algorithms with larger compression gains will be possible. Information sources and sinks are also required to know super coefficient matrices for each of the considered Galois fields. Even a large $1000x1000$ coefficient matrix which has elements of Galois field 2^8 will only occupy 0.95MB. Therefore, storage of super coefficient matrices is not a challenge. Furthermore, it should be noted that intermediate nodes are not required to possess knowledge about the super coefficient matrices. Thus, intermediate nodes are not able to decode source information. Hence, the usage of the proposed method can also be extended to increase data security. Furthermore, as coefficient information includes information on how source packets are combined throughout the network, collecting coefficient information from different paths can be used for network tomography.

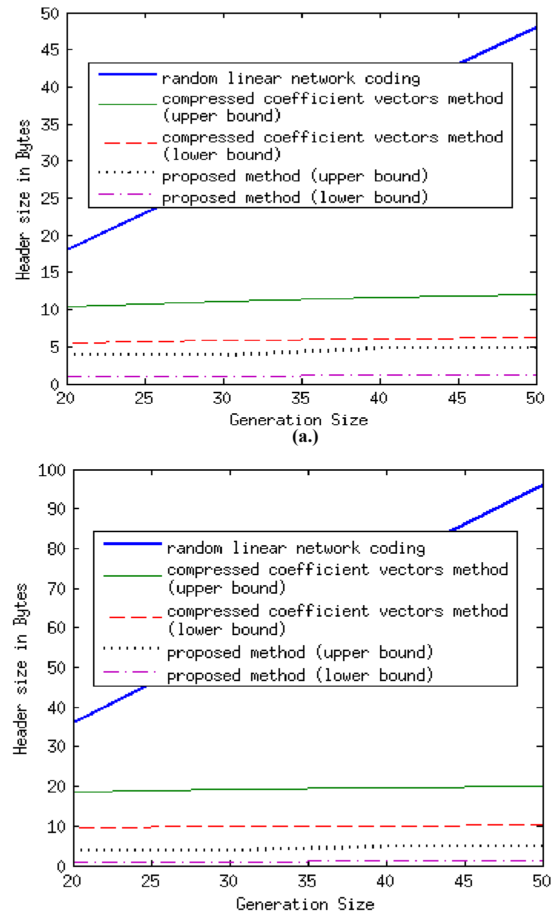

Fig. 2. Header sizes in bytes vs. number of packets in generation for Galois fields (a.) 2^8 and (a.) 2^{16}.

4. CONCLUSIONS

This paper presents a novel approach to minimise the coefficient vector size of network coded packets in small and medium sized networks. Authors exploit the facts that the number of source packets that get combined are limited and k < F number of bits are sufficient to represent each element of coefficient information. Simulation results exhibit that the proposed method has a better capability to compress network coding coefficient vectors compared to existing algorithms. As for future work, the proposed approach of using coefficient information to compress coefficient vectors can be extended to be used for large scale complex network structures.

Acknowledgement: This work has been supported by the MUSCADE Integrating Project (www.muscade.eu), funded under the European Commission ICT 7th Framework Programme.

5. REFERENCES

[1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, “Network information flow,” Information Theory, IEEE Transactions on, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] R. Koetter and M. Medard, “An algebraic approach to network coding,” IEEE/ACM Trans. Networking, vol. 11, no. 5, pp. 782–795, October 2003.

[3] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, “The benefits of coding over routing in a randomized setting,” in Information Theory, Proceedings. IEEE International Symposium on, pp. 442, 29 June-4 July 2003.

[4] P. A. Chou, Y. Wu, and K. Jain, “Practical Network Coding,” in Annual Allerton Conference on Communication Control and Computing, vol. 41, pp. 40–49, 2003.

[5] R. Koetter and F. R. Kschischang, “Coding for errors and erasures in random network coding,” Information Theory, IEEE Transactions on, vol. 54, no. 8, pp. 3579–3591, 2008.

[6] M. Jafari, S. Mohajer, C. Fragouli, and S. Diggavi, “On the capacity of non-coherent network coding,” in Information Theory, 2009. ISIT 2009. IEEE International Symposium on, pp. 273–277, 2009.

[7] M. Jafari, L. Keller, C. Fragouli, and K. Argyraki, “Compressed network coding vectors,” in Information Theory, 2009. ISIT 2009. IEEE International Symposium on, 2009, pp. 109–113, June 28-July 3 2009.

[8] "Internet2 Network." Internet: http://www.internet2.edu/.

[9] "Janet." Internet: https://www.ja.net/.

[10] "Verizon." Internet: http://www.verizonbusiness.com/uk/.