RESEARCH ARTICLE

Minimal Increase Network Coding for Dynamic Networks

Guoyin Zhang[✉], Xu Fan^{*✉}, Yanxia Wu

College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang, P. R. China

✉ These authors contributed equally to this work.

* fanxu@hrbeu.edu.cn

OPEN ACCESS

Citation: Zhang G, Fan X, Wu Y (2016) Minimal Increase Network Coding for Dynamic Networks. PLoS ONE 11(2): e0148725. doi:10.1371/journal.pone.0148725

Editor: Peter Csermely, Semmelweis University, HUNGARY

Received: September 9, 2015

Accepted: January 22, 2016

Published: February 11, 2016

Copyright: © 2016 Zhang et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work was supported by No. 61073042 and No. 61170241, <http://www.nsf.gov.cn/>, National Natural Science Foundation of China, GZ. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

Because of the mobility, computing power and changeable topology of dynamic networks, it is difficult for random linear network coding (RLNC) in static networks to satisfy the requirements of dynamic networks. To alleviate this problem, a minimal increase network coding (MINC) algorithm is proposed. By identifying the nonzero elements of an encoding vector, it selects blocks to be encoded on the basis of relationship between the nonzero elements that the controls changes in the degrees of the blocks; then, the encoding time is shortened in a dynamic network. The results of simulations show that, compared with existing encoding algorithms, the MINC algorithm provides reduced computational complexity of encoding and an increased probability of delivery.

Introduction

Network coding, which was first proposed in the context of information theory, has been introduced into communication networks to improve transmission efficiency and to ease block propagation scheduling [1, 2]. It has been shown that network coding enables a wide variety of applications, including communications in ad-hoc networks and data collection in sensor networks, by exploring the broadcast essence of wireless networks [3, 4]. Since the block is the universal component of network coding, a node can transmit any encoded block to other nodes, and each block has a high probability of equal contribution to the nodes.

RLNC [5] has been proven to be a powerful tool for disseminating information over a network. According to the previous studies of RLNC [2, 5], encoding operations are performed by both the sources and the intermediate nodes. Whenever a node or a source needs to forward a block to another node, it simultaneously produces a linear combination of all of the blocks in storage.

Even though RLNC greatly facilitates network implementation by providing a decentralized solution, the relatively high complexity of encoding at sources and intermediate nodes remains a major obstacle to its widespread use. To reduce the complexity of the coding process, Chou et al. [6] proposed the concept of group network coding, in which blocks are encoded within the same group or segment, with each segment containing a prescribed

number of blocks. This coding scheme, i.e. using all of the plain blocks in a given segment to generate encoded blocks, is very highly complex and requires a great deal of overhead for disk operation. The sparse form of RLNC (Paircoding), in which only two random plain blocks are used to generate new blocks, is proposed in [7]. Paircoding is regarded as at least as good as BitTorrent for file sharing, and the two have nearly the same computational complexity in terms of disk access. However, Paircoding cannot be fully exploited because of the low rate at which it generates new blocks. Cai et al. [8] analyzed the potential for improvement in the content made available and bandwidth used by two existing network coding schemes and proposed a sparse network coding scheme to accelerate the generation of new blocks. From the perspective of group network coding, each segment in this coding scheme contains two plain blocks. Niu and Li [9] quantitatively evaluated how network coding can be used to improve content availability, block diversity, and download performance in the presence of churning as the number of blocks varies in each coding segment. To reduce the complexity of decoding, Feizi et al. [10] noted the need for nodes in different transmission phases and designed a tunable sparse network coding algorithm in which the density of the network coded packets varies during a transmission. An encoding window [11] is an effective tool for controlling the computational complexity of an encoding operation that encodes only the data that have a nonzero component within the window. As Paircoding does, an encoding window reduces the computational complexity while limiting the generation speed of encoding a block.

We revisit the idea of using sparse coding to reduce the complexity of the encoding process via a fundamentally different approach in which a newly generated block is deployed to maintain a minimal difference from the existing blocks. This scheme is named MINC. The remainder of this paper is organized as follows: In Section 2, a brief introduction is offered to the basic mathematical description of RLNC. Section 3 will witness our proposal of the MINC scheme and the analysis of its probabilistic properties. Section 4 is spared for an experimental evaluation of the encoding speed and distribution efficiency of MINC. Finally, Section 5 concludes the whole paper with a summary of the research findings.

Model and Preliminaries

In this section the principles will be delineated behind RLNC. Then, it will be demonstrated how random packets get recombined at the nodes of a randomly connected graph. The following part takes into account of a general multicast network with one source and several receivers. Then the network will be modeled as a directed graph, $G = (V, E)$, where V is the set of nodes in the network and E is the set of edges in G , E is a subset of $V \times V$ which represent point-to-point channels.

The source node holds a message, F , i.e. a generated file, that must be distributed to multiple network nodes. In RLNC, the file is divided into G segments, each of which is further broken into m blocks in the source node. If segment i has an original block, $\mathbf{B}^i = [B_1^i \ B_2^i \ \cdots \ B_m^i]^T$, then, an encoded block, b^i , from segment i is a linear combination of $[B_1^i \ B_2^i \ \cdots \ B_m^i]^T$ in

the Galois field $GF(q)$, i.e. $b^i = \sum_{j=1}^k c_j B_j^i \ 1 \leq j \leq m, c_j \in GF(q)$.

If a node has $l (l \leq m)$ encoded or un-encoded blocks of segment i , $\mathbf{b} = [b_1^i \ b_2^i \ \cdots \ b_l^i]^T$, then, when serving another node, p , it randomly selects a set of coding coefficients from $GF(q)$, namely, the coding vector denoted by $\mathbf{c} = [c_1 \ c_2 \ \cdots \ c_l]$, and then, encodes all of the blocks in seg-

ment i and produces one encoded block, $x^i = \sum_{j=1}^l c_j \cdot b_j^i = \mathbf{c} \cdot \mathbf{b}$. The encoding vector of the

new block is denoted by $\mathbf{y}$, where $\mathbf{y} = (y_1 y_2 \cdots y_n) = (c_1 c_2 \cdots c_n)$. We use the encoding vector to represent the corresponding block in the following sections.

In an encoded block, x , the coding coefficients used to encode original blocks are embedded in the header of the encoded block. As soon as a peer, p , successfully receives a total of m encoded blocks from segment i , $\mathbf{x} = [x_1^i \ x_2^i \ \cdots \ x_m^i]^T$, that are linearly independent, it is able to decode segment i with the coding coefficients embedded in each of the m encoded blocks. To decode segment i , we must first compute the inverse of the coefficient matrix $\mathbf{A}_i$ by Gaussian elimination. To obtain the original m blocks, it is then necessary to multiply $\mathbf{A}_i^{-1}$ and $\mathbf{x}$, i.e. $\mathbf{B}^i = \mathbf{A}_i^{-1} \times \mathbf{x}$.

Minimal Increase Network Coding

As described in the section on the coding process, we know that RLNC adopts a high-density encoding scheme to ensure that the encoding vectors are linearly independent, which leads to excessive computational overhead during the encoding process. From the perspective of decoding, the relationship among the encoding vectors, not the coding density, determines whether a receiver can decode a source file.

The nonzero elements of an encoding vector are denoted by the set Z_y , i.e. $Z_y = \{x | y_x \neq 0\}$. For any two encoding vectors $\mathbf{y}$ and $\mathbf{y}'$, if $Z_y \setminus Z_{y'} \neq \emptyset$, we say that $\mathbf{y}$ is not less than $\mathbf{y}'$ and write $\mathbf{y} \triangleright \mathbf{y}'$. Intuitively, $\mathbf{y} \triangleright \mathbf{y}'$ explains that $\mathbf{y}$ contains nonzero elements that differ from those of $\mathbf{y}'$. The term “degree” is employed to represent the number of nonzero elements in an encoding vector; the degree of an encoding vector is represented by $d(\mathbf{y})$. When $\mathbf{y} = 0$, $d(\mathbf{y}) = 0$. The average degree of the encoding vectors in node v is denoted by $\bar{d}$. To facilitate processing, we add Z_y to the area in the encoded packet; the packet format is shown in Fig 1.

A source node, S , randomly selects two plain blocks for each encoding operation, and the encoding coefficients are selected from $GF(2^q) \setminus \{0\}$. The intermediate nodes control the process of choosing which block to encode based on its degree and relationship to Z_y . One difference between the MINC and the RLNC methods is that in the former, node v saves the encoded block as a benchmark after transmitting it. The encoding vector of a benchmark block is denoted by $\mathbf{y}^0$, and the degree of $\mathbf{y}^0$ is denoted by d_0 . Before encoding the next block, node v selects a block that satisfies the condition $\mathbf{y}' \triangleright \mathbf{y}^0$ to be encoded. The encoding coefficients are selected from $GF(2^q) \setminus \{0\}$ by the source node. Then, the benchmark block is updated with the help of the newly encoded block. When $d_0 > \bar{d}$, node v selects a block that satisfies $d \leq \bar{d}$ as the new benchmark block. The proposed encoding algorithm is shown in Algorithm 1.

Algorithm 1. Minimal Increase Network Coding for Intermediate Nodes.

Input: $\mathbf{y}^i$ ($1 \leq i \leq k$), $\mathbf{y}^0$
Output: $\mathbf{y}'$

```

1: if ( $d(\mathbf{y}^0) = 0$  or  $d(\mathbf{y}^0) > \bar{d}$ ) then
2:    $\mathbf{y}^0 \leftarrow \mathbf{y}^i$  of  $d(\mathbf{y}^i) \leq \bar{d}$ 
3: end if
4: while  $\mathbf{y}^i \not\triangleright \mathbf{y}^0$  do
5:    $i++$ 
6: end while
7:  $\mathbf{y}' \leftarrow c_i \mathbf{y}^i + c_0 \mathbf{y}^0$ 
8:  $\mathbf{y}^0 \leftarrow \mathbf{y}'$ 

```

Since it is necessary to read in the data and perform multiplication and addition in a finite field, the encoding operation is responsible for most of the algorithm’s calculations. Step 9 in Algorithm 1 is where the encoding is performed; only two data blocks need to be encoded during this process. The total overhead involved in coding is proportional to the number of data blocks; therefore, MINC requires less computation for the encoding process than RLNC does.

head	encoding vector	Z_y	data
$hhh...h$	$y_1y_2...y_n$	$\{x y_x \neq 0\}$	Data

Fig 1. The MINC data packet format.

doi:10.1371/journal.pone.0148725.g001

Next, we analyze the mathematical properties of MINC. Since the exact characterization of network coding in MP2P network is extremely difficult to be obtained, approximated analysis is applied onto such protocols based on several assumptions. In every discrete time-step, each node selects a limited communication partners randomly but uniformly among all nodes with only one message transmitted [12]. Thus the following assumptions are maintained: (1) The number of data blocks received by an intermediate node satisfies a Poisson distribution with parameter λ . (2) The non-zero elements of a coding vector received by node v are uniformly distributed.

At time t , node v has λt blocks in its buffer; their encoding vectors are $d_1, d_2, \dots, d_{\lambda t}$. $\bar{d} = \frac{\sum_{i=1}^{\lambda t} d_i}{\lambda t}$ denotes the average degree of all of the blocks in node v . Under the given conditions, the probability that node v selects y' from λt blocks and $y' \triangleright y^0$ is $P = 1 - \left(1 - \frac{\sum d_i}{\lambda t \cdot n}\right)^{n-\bar{d}}$.

Theorem 1: The expectation of the number of times that a node first gets eligible block is less than $\frac{1}{P}$.

Proof: Denoted the number of times that a node first gets the eligible by X . It is easy to see that X is maximized when the degree of the benchmark block is $\bar{d}$. Because X has a geometric distribution with parameter P , the expectation is $E_1(X) = \sum_{j=1}^{\infty} j \cdot P \cdot (1-P)^{j-1} = \frac{1}{P}$.

Theorem 2: The expectation of the degree of a new encoded block is less than $nP + (1-P)\bar{d}$.

Proof: According to the process of encoding with MINC, the degree of a generated block is maximized when the degree of the benchmark block is $\bar{d}$. Therefore, the expectation of the maximum degree is $E_2 = \bar{d} + E'$, where E' is the expectation of an increased degree when $d_0 = \bar{d}$. The increased degree has a binomial distribution with parameters $(n - \bar{d}, P)$ when $d_0 = \bar{d}$; therefore, $E' = kC_{n-\bar{d}}^k P^k (1-P)^{n-\bar{d}-k} = (n - \bar{d})P$, $E_2 = nP + (1-P)\bar{d}$.

Performance Evaluation

In this section, performance evaluation of RLNC, sparse NC and MINC encoding algorithms is launched comparatively for different conditions. To be specific, the algorithms are evaluated for their encoding speed, data distribution speed in dynamic networks. To evaluate the performance of the MINC algorithm, a simulator is constructed that afford the observation and measurement of the algorithm performance under a variety of conditions. The basic simulation

parameters are chosen to model a dynamic network consisting of a collection of mobile nodes moving in an area with a fixed size.

The nodes' motion is modeled with the modified random waypoint model [13]. All simulations are carried out in a rectangular area of 1500m*500m, the results of which are averaged for at least 100 simulation runs. Besides, node speeds are randomly distributed between 2m/s and 15m/s. At the start of every simulation, each node is initially placed at a random position within the simulation area and randomly chooses a new location as its destination. After reaching the destination, the node waits for a given pause time and starts moving towards some other randomly chosen destination.

To directly compare the performances of the coding algorithms on the basis of experimental results, a communication channel is configured with a high rate and uniform connection delay, with the influences of other factors ignored on the underlying network in that the simulation is for encoding performance only, not for a thorough performance comparison. Other specific experimental parameters settings are shown in Table 1.

Comparison of the encoding speeds

The basic idea runs as follows. The total time required is recorded to generate a certain number of encoded blocks of different sizes using the MINC and RLNC. The time required is recorded after the 200 MB encoded block has been generated using RLNC and MINC, and then, the average encoding speed is calculated. The results are shown in Fig 2.

From the results shown in Fig 2, it is clear that MINC is faster than RLNC. The maximum encoding speed of the MINC algorithm exceeds 4.5 MB/s, which satisfies the requirements of general dynamic network transmission. In the same hardware environment and with the same data block size, the encoding speed of the MINC algorithm is significantly greater than that of the RLNC algorithm. In addition, as the figure shows, the encoding speeds have the same trend with respect to the block size; as the block size increases, the number of blocks decreases, which reduces the amount of switching between selecting coefficients and encoding, therefore, increases the encoding speed. However, the curves have inflection points when the block size is 256 KB, which may be due to the system's cache size, access control, or other factors.

Comparison of the data distribution speeds

The speed of data distribution is an important index of an algorithm's performance. In addition to the encoding speed, the structure of an encoded block influences the velocity distribution, which is primarily shown in the linear correlations between blocks generated with different algorithms. A node encodes all of the data blocks in its domain of RLNC, which minimizes the probability of linear dependence; in contrast, a sparse encoding scheme increases the probability because it reduces the amount of data involved in the encoding process. In a comparative experiment, the file size is set to 100 MB and the block size to 256 KB and the results

Table 1. The parameters used in the simulations.

parameter	value
file size	100,200MB
block size	64-2048KB
nodes	50
communication distance	100m
pause time	0-3s

doi:10.1371/journal.pone.0148725.t001

Fig 2. A comparison of the encoding speeds.

doi:10.1371/journal.pone.0148725.g002

are compared between RLNC, 8-sparse NC, 4-sparse NC and MINC. The results are shown in Fig 3.

In Fig 3, the curves can be divided into three phases. In the first phase, zero or only a small number of data blocks are cached in nodes; therefore, data distribution begins slowly. In the second phase, which is the data distribution phase, most of contents are distributed. In the third phase, the target node receives a large amount of data, which increases the probability of linear dependence between new blocks and other blocks, yielding the decrease in distribution speed. In the second phase, the encoding speed is the primary factor influencing the distribution speed. The encoding speed of MINC supports the distribution speed more strongly than that of RLNC; 4-sparse and 8-sparse NC also perform well. The differences in the performances of the algorithms are very apparent in the third phase; the speed of the MINC algorithm decreases slightly, that of the RLNC algorithm is essentially stable, and that of fixed density encoding decreases significantly, that of especially 4-sparse NC. The experimental data reveal that the MINC algorithm distributes data approximately 7% faster than it encodes data completely.

To study the effects of different block sizes on the distribution speed, a set of experiments is conducted on a 100 M file with a block size between 64 KB and 2048 KB. The results are shown in Fig 4.

From Fig 4, it is shown that the curves corresponding to MINC and RLNC are similar to the "V" glyph when the block size increases, which betrays that the speeds of the two algorithms affected the encoding speed and the structure of the encoding vector. The total distribution times of 4-sparse NC and 8-sparse NC tend to increase because the number of blocks involved in encoding is fixed; therefore, the speed is primarily affected by the construction of the

Fig 3. A comparison of the data distribution speeds.

doi:10.1371/journal.pone.0148725.g003

Fig 4. The effect of the block size on the distribution speed.

doi:10.1371/journal.pone.0148725.g004

encoding vector. When the total number of blocks is reduced, the probability of linear dependence between blocks increases, and so does the download time.

From these results above, it can be concluded that MINC performs better for various block sizes in the modified random waypoint model. Therefore, MINC is better suited to a dynamic network environment than other encoding algorithms.

Conclusions

Although there are numerous encoding algorithms for network coding, many of them suffer from computational complexity or a lack of distribution efficiency. This paper aims to investigate the feasibility of reducing the computational complexity. To leverage the potential of network coding and keep the computational complexity low, a MINC scheme is proposed in which the encoder selects blocks to be encoded on the basis of the nonzero elements of the encoding vector. Since only two blocks are involved in the generation of new blocks, the computational complexity remains low. The effectiveness of the proposed coding scheme on the computational complexity in a dynamic network is demonstrated by means of simulations and performance comparisons. In the tradeoff between computation complexity and file sharing effectiveness, MINC ranks as an excellent compromise among the other schemes.

Supporting Information

S1 Fig. A comparison of the encoding speeds.

(TXT)

S2 Fig. A comparison of the data distribution speeds.

(TXT)

S3 Fig. The effect of the block size on the distribution speed.

(TXT)

Acknowledgments

We would like to acknowledge the suggestions of Dr. Xinyu Zhang for performance evaluation.

Author Contributions

Conceived and designed the experiments: GZ XF. Performed the experiments: XF YW. Analyzed the data: XF YW. Contributed reagents/materials/analysis tools: GZ XF. Wrote the paper: XF. Mathematical proof: XF.

References

1. Ahlswede R, Ning Cai, Li S-YR, Yeung RW. Network information flow. *IEEE Transactions on Information Theory*. 2000; 4(46), 1204–1216.
2. Gkantsidis C, Rodriguez PR. Network coding for large scale content distribution. *INFOCOM 2005*. 2005; 3: 2235–2245.
3. Akhtar AM, Nakhai MR, Aghvami AH. On the Use of Cooperative Physical Layer Network Coding for Energy Efficient Routing. *IEEE Transactions on in Communications*. 2013; 4(61), 1498–1509.
4. Rout RR, Ghosh SK. Enhancement of Lifetime using Duty Cycle and Network Coding in Wireless Sensor Networks. *IEEE Transactions on in Wireless Communications*. 2013; 2(12), 656–667.
5. P Chou, Y Wu, and K Jain. Practical Network Coding. in *Proceedings of Allerton Conference on Communication, Control and Computing*, October 2003.
6. Ho Tracey, Medard M, Koetter R, Karger DR, Effros M, Jun Shi, et al. A Random Linear Network Coding Approach to Multicast. *IEEE Transactions on Information Theory*. 2006; 10(52), 4413–4430.

7. Ortol C, Schindelhauer C, Vater A. Paircoding: Improving File Sharing Using Sparse Network Codes. *Internet and Web Applications and Services* 2009. 2009; 24–28.
8. Cai QingChao, Lo KwokTung. Two Blocks Are Enough: On the Feasibility of Using Network Coding to Ameliorate the Content Availability of BitTorrent Swarms. *IEEE Transactions on Parallel and Distributed Systems*. 2013; 8(24) 1682–1694.
9. Niu Di, Li Baochun. Analyzing the Resilience-Complexity Tradeoff of Network Coding in Dynamic P2P Networks. *IEEE Transactions on Parallel and Distributed Systems*. 2011; 11(22), 1842–1850.
10. Feizi S, Lucani DE, Sorensen CW, Makhdoumi A, Medard M. Tunable sparse network coding for multicast networks. *2014 International Symposium on Network Coding (NetCod)*. 2014; June, 1–6.
11. Fiandrotti A, Bioglio V, Grangetto M, Gaeta R, Magli E. Band Codes for Energy-Efficient Network Coding With Application to P2P Mobile Streaming. *IEEE Transactions on Multimedia*. 2014; 2(16), 521–532.
12. Deb S, Medard M, Choute C. Algebraic gossip: a network coding approach to optimal multiple rumor mongering. *IEEE Transactions on Information Theory*. 2006; 6(52), 2486–2507.
13. Yoon J, Liu M, Noble B. Random waypoint considered harmful. *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications in INFOCOM* 2003. 2003; vol.2, 1312–1321.