Sparse Network Coding Based on N-dimensional Maximal Independent Set

Dongqiu Zhang[1,*], Chunhua Ma[2], Jishan Zhang[2], and Xiaoxia Sun[2]

[1]Computer Science Dept. Mudanjiang College, Mudanjiang, 157001, China;

[2]Information Engineering Dept. Suihua University, Suihua, 152000, China.

*307642064@qq.com

Abstract

In [1], a scenario is presented where a source node s wants to reliably distribute messages to many one-hop sink nodes. The channels are erasure channels. We present a new distribution scheme based on our proposed novel mathematical concept of n-dimensional maximal independent set(nMIS). It is called sparse network coding based on n-dimensional maximal independent set for one-hop broadcasting networks (SNCnMIS). Generally speaking, the source needs to send $n+k$ packets when any sink needs n packets because of bit error. This scheme makes every n out of $n+k$ coding vectors are linear independent in n-dimensional vector space. In essence, it makes the possibility, that the coding vector of n received packets are dependent, to be zero. It is also a sparse coding scheme which reduces the computational complexity of encoding and decoding. Obviously, this scheme sends the least packets, then, it can get good performance in the gain of the delay, energy and throughput.

Keywords

Maximal independent set, sparse network coding , linear independent.

1. Introduction

Network coding can bring great advantages in the broadcasting network. Many network coding schemes have been presented to such environments as one-hop broadcasting network. [2] gets statistics of shared lost packets of some sinks, and informs the source the statistics through ACKs. Then the source retransmits the shared packets. Obviously, this scheme reduces the retransmitted packets because many sinks share the same lost packets. But many retransmissions are also adopted here, and they result in more delay and energy cost. Can we reduce the retransmissions as much as possible? It would be natural to think that the source must ensure the validity of the packets to send. That means that the sink should reach a high probability of generating independent encoded packets every time. Because of bit error, the source needs to send $n+k$ packets when any sink need n packets because k packets will lost. Assuming that it is error-free channel, the "independent" means n linear independent packets are sent if we want to send n packets in network coding scheme. All the packets are received without loss. But in the erasure channels, k redundancy packets are needed for transmitting n packets. In this context, what is the metric of "independent"? Intuitively, we can ensure that n packets out $n+k$ are linear independent. The so-called systematic coding in [1] has done just like that. Compared with the scheme of randomly selecting the coding coefficients, the method just makes limited benefit as the k becomes bigger. The reason is that, at the beginning of encoding, though we can choose n of $n+k$ packets are independent, k packets are still dependent with the n packets. If the lost packets are among the selected n original independent packets, the received n packets are surely dependent. Can we improve it? May be we can increase the independent possibility of every n packets out of $n+k$.

There are two methods to ensure that among pre-existing methods. First, to perform a completely random coding in a big finite field. It is also a conventional method. Not considering of packets loss, the expected average transmitted packets are $n*q/(q-1)$ if n original packets are aimed to be received in the sink where q is the size of field [1]. Obviously, if q is big enough, the number of redundancy packets k can be very small. But big filed leads to the increase of the computational complexity[3]. The second method is the scheme as sparse network coding in [4] and LT fountain codes in [5]. The essence of them is to encode original packets according to a good probability distribution which can ensure the coded packets have a high independent possibility. The processes are both performed in a desired small field. So they both also have lower computational complexity.

All the schemes are to ensure a high independent possibility of coded packets as much as possible. What is the optimal conditions in the context that $n+k$ are sent where k packets will lost because of bit error? In this paper, we propose a scheme to ensure that every n packets out of $n+k$ are linear independent in a small field to reach the optimal solution. Just like [3], we also consider the computational complexity. n systematic codes, i.e., the coding vectors are n standard basis, are included in $n+k$ packets. The systematic codes and the small field ensure a low computational complexity. Obviously, this also results in less retransmissions. For a sink, the information loss are caused by two reasons. First reason is the channel noise, i.e., k packets out $n+k$ will lost if the packets lose rate is $k/(n+k)$. This is a objective existence, and we have to consume redundancy packets to counteract the channel noise. Another reason is the received packets are independent. The smaller the rank of coding vectors, the more the information loss. If the packets lose rate is $k/(n+k)$, and every n packets out of n+k are linear independent, the scheme will reach the optimal solution. The scheme is based on our proposed novel mathematical concept of n-dimensional maximal independent set(nMIS).

The main contributions of this letter are as follows:

a novel mathematical concept of n-dimensional maximal independent set(nMIS) is proposed

a network scheme SNCnMIS is proposed. This scheme achieves the theoretically optimal performance in reducing the possibility of dependent packets.

this scheme adopts standard basis of the vector space as the coding vector which reduces the complexity of both coding and decoding processes. It also performs in a small field which reduces the computational complexity further.

a algorithm to get the nMIS for certain dimension n and the size of field q is proposed.

2. N-dimensional Maximal Independent Set

As a theoretical support, the concept of nMIS is necessary to be illustrated here. This idea is mainly inspired by [6] where 2-dimensional maximal independent set is presented. We generate this concept to arbitrary dimension vector space.

Definition 1: A set of $n+k$ linear vectors $v_1,...,v_{n+k}$ is called a n-dimensional independent set, if every n $(0 \le k)$ vectors are linear independent. A set is called a n-dimensional maximal independent set(nMIS), if $n+k$ is maximal.

Denote the cardinality of nMIS over a finite field with Cline. What is the key question about what happens next? It is certainly how to get the nMIS for certain n and q. Is there an algorithm with polynomial complexity to get it? It is with regret that we have not found a quick algorithm except the exhaust algorithm so far. But this will not degrade the performance of SNCnMIS. We can compute the nMIS with high performance computer for some n and q in advance. Then the nMIS can be shared as the code book in the internet. If the nMIS is got in the source in advance, the encoding complexity will be as that of usual coding. Another concern involves that small n and q are inefficient to perform network coding. But in [7] it shows even small n and q(q=4) may also effective to carry out network coding.

Table 1. The Cardinality of nMIS Cline

q \ n	2	3	4	5	6	7
2	3	3	3	3	3	3
4	6	7	7	7	7	7
8	14	12	11	10	10	10
16	30	25	17	14	10	

So far, we have got the nMIS for some n and q in the common PC. We have not yet got the nMIS for the blank form in Table 1.

Assume $n+k=C_{line}$. Generally speaking, k is inversely proportional to n, and is direct proportional to q. But that is just a empirical observations more than a rigorous theorem. For example, with common PC, to computer the nMIS, it costs nearly 2 hours for n=4, q=16. It is 7 minutes for n=6, q=8. For example, when n=2, q=2, nMIS=[0 1;1 0;1 1]. when n=3, q=4, nMIS=[0 0 1;0 1 0; 1 0 0;1 1 1;1 2 3;1 3 2; 2 1 3].

The algorithm to get a nMIS is as follows. The algorithm is recursive and exhaust. The main body includes two sub algorithms.

Subalgorithm1(matrix1, matrix2) is to get the set of vectors T in matrix2, and the juxtaposition of every one of T and every n-1 vectors of matrix1 are full rank. Subalgorithm1 is easy and we omit the pseudo-code. Subalgorithm2 is as follows.

```
Procedure  P=Subalgorithm2(matrix3, matrix4)
//to get the maximal sub-matrix P of matrix4. //Every n vectors of the matrix, which is the //juxtaposition of //matrix P and matrix3, are //independent.
MIS= matrix3
M= Subalgorithm1(matrix3, matrix4)
if  ~isempty(M)          //if M is not empty
P=[]             //P is empty set
else
P= M(1,:)      //make the first row //vector of matrix M as one //vector of P.
MIS=[MIS; M(1,:)] //update MIS as the first //parameter of //Subalgorithm2(MIS, M) by //juxtaposing MIS and M(1,:)
end
P=[P; Subalgorithm2(MIS, M)]
//recursive function.
//two parameters is updated to MIS //and M
end procedure
```

The ultimate algorithm pseudo-code is getnMIS(n,q).

```
Procedure  Q=getnMIS(n,q)
//to get the ultimate nMIS based on n and q.
//V(n,q) to get the n-dimensional vector //space including qn vectors.
base (n)
//to get n standard basis
P'=subalgorithm2(base (n), V(n,q) )
Q= [base (n);P']
//juxtaposes standard basis and P', and Q is //the ultimate nMIS.
End procedure
```

3. Coding Schemes

If the nMIS is got in the source in advance, the encoding complexity is common with usual network coding. The encoding process is as follows. Denote the n standard basis vectors with N, and denote the k non-standard basis vectors with K where $N \bigcup K = C_{line}$.

Procedure encoding

Divide the original messages into n packets;

Send n packets out without encoding;

if (have not received the all the ACKs from all the sinks)

send the message encoded with one vector of K and

the vectors of K is selected one by one in cycle mode;

else

the encoding process is over;

end

End procedure

The decoding process is as follows;

Procedure decoding

With received original packets replace the variables of other received packets whose encoding vector is not standard basis;

With Gaussian elimination method to solve the left equations；

End procedure

A few points are needed to note. They are as follows.

Based on whether the bit error rate p_b is a statistical terminology, the method to calculate the transmission efficiency is different. Denote the number of additional packets needed to be sent is f, and f is to ensure that all the n original packets can be decoded. Assume the transmission efficiency is that the number of additional packets needed to be sent f divides the number of original packets n, i.e., f / n.

If p_b is assumed to be a constant, we will calculate the transmission efficiency as follows. we will evaluate the bit error rate of the channel. Then the desired value of n and q are selected and the size of one packet L is confirmed. Based on n, the size of one packet and the bit error rate determine the number of average lost packet, namely, k. We must ensure $n+k \le C_{line}$. If that, after C_{line} coded packets are sent, all the sinks can decode the original packets. If the packet size is L bits, $P = 1-(1-p_b)^L$ where P is the packet loss rate. k is the number of average lost packets. Then, $k = n * P$. For simplicity, assume the number of additional packets needed to be sent f is error-free. Obviously, $f = k$ because p_b is a constant. In this case, the transmission efficiency is k / n.

But in a real environment, p_b is a statistical terminology because of data modulation type, unstable physical environment. In this case, we will calculate the number of the additional packets f with another way. Because the bit error rate p_b is a statistical terminology, P and k may deviate from theoretical value with a small probability . So, after $n+k$ coded packets are sent, the ACKs may not come with a small probability. This time, $k = n * \mathrm{E}(P)$ where $\mathrm{E}(P)$ is the expected value of P. So, the set of vectors K with cardinality k is selected one by one in cycle mode in encoding scheme just in case. After sending the n original packets, the source will not stop to send the packets among the set of nMIS excepts the n original messages until it receives all the ACKs. From this perspective, it is also a rateless coding.

How to calculate the number of additional packets needed to be sent f in this situation? As a preface, random linear network coding is considered. If random linear network coding is used, the expected number of success fully received packets before having n linearly independent combinations is $\sum_{j=1}^{n}\frac{1}{1-(1/q)^j}\le n\frac{q}{q-1}$ [8]. Including the average lost k packets, the number of all the sent packets is approximate $nq/(q-1)+k$. In SNCnMIS, if p_b is a constant, the number of all the needed packets is $n+k$, i.e., $n+n*E(P)$. By now, $C_{line} > n+k$. If P is accord with a probability distribution, the expected number of all the sent packets is like equation (1).

$$\begin{aligned} n+f &= n+\int_{-\infty}^{C_{line}} f(x)dx + \frac{q}{q-1}\int_{C_{line}}^{+\infty} f(x)dx \\ &= n+\int_{-\infty}^{+\infty} f(x)dx + \frac{1}{q-1}\int_{C_{line}}^{+\infty} f(x)dx \\ &= n+n*E(P) + \frac{1}{q-1}\int_{C_{line}}^{+\infty} f(x)dx \\ &= n+k+\frac{1}{q-1}\int_{C_{line}}^{+\infty} f(x)dx \end{aligned} \tag{1}$$

$f(x)$ is the probability density function. If C_{line} is much larger than k, $\int_{C_{line}}^{+\infty} f(x)dx$ can be ignored. This time, the normalized transmission efficiency is $(k+1/(q-1)*\int_{C_{line}}^{+\infty} f(x)dx)/n$. It also can be expressed by $E(P)+1/(q-1)*(\int_{C_{line}}^{+\infty} f(x)dx)/n$. Obviously, the normalized transmission efficiency in this case is worse than that in the case p_b is assumed to be a constant.

N is the set of the standard basis vector with size n, and coded with them, n original packets are sent without too much computational work. When decoding, the n original packets reduce the matrix just to nearly at most k dimension matrix. This saves a large amount of computing resources. So, like [4]-[5], our scheme is also a sparse coding.

4. The experiment and discussion

Assume p_b is accord with normal distribution with the mean $E(p_b)=0.0005$. It is a high bit error rate, and is more common in underwater environment. Firstly, the packet length is set to 20 bytes. Because the adopted n and q is small, the overhead from the coding vector is omitted for simplicity. The mean is $E(P)=0.01$ based on the equation $P=1-(1-p_b)^L$. Assume the variance is $\sigma=3$, then the packet loss rate is accord with normal distribution with the mean $E(P)=\mu=0.01$ and the variance σ. σ is mainly determined by data modulation type, physical environment and so on. σ becomes bigger when the environment becomes hash. If P is accord with normal distribution, i.e., $P\sim N(\mu,\sigma^2)=N(0.01*n,\sigma^2)=N(k,\sigma^2)$.

In Fig. 1, L=20. The key of algorithm SNCnMIS is to ensure that $C_{line}>k$. If p_b is a constant, the transmission efficiency is $k/n=0.01$. If p_b is vary with time, the more the difference between C_{line} and k, the more efficiency the SNCnMIS. So, as the growth of q, the efficiency of SNCnMIS is getting better because the value $C_{line}-k-n$ becomes bigger. From Fig. 1, we know that the transmission efficiency is the best when q is 16. The reason is that $C_{line}-k-n$ has maximum value. If random linear network coding is used, the transmission efficiency is $q/(q-1)+k/n$. So, RLNC has poor performance.

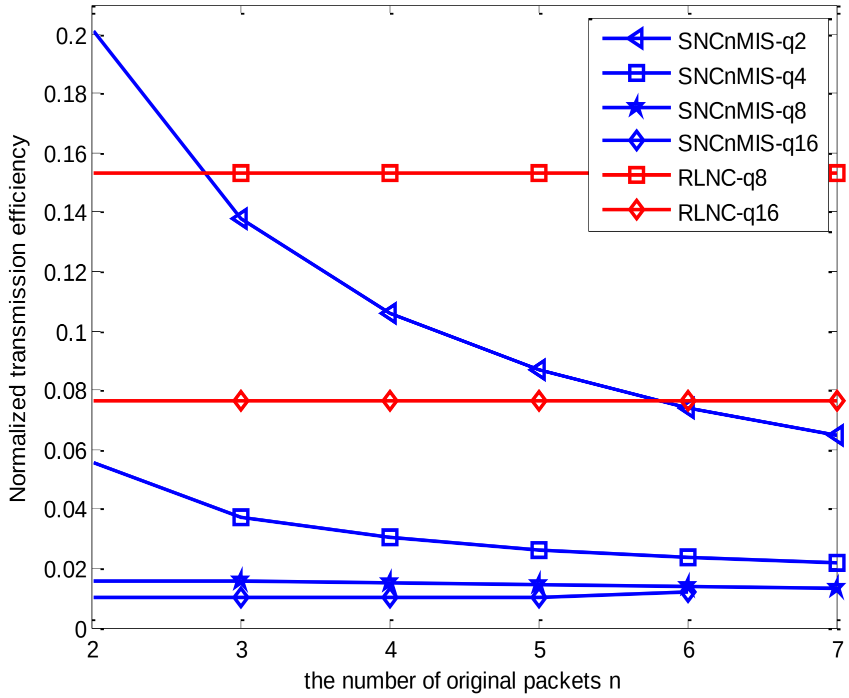

Fig. 1. The Transmission Efficiency of RLNC and SNCnMIS When L=20 Bytes and σ=3.

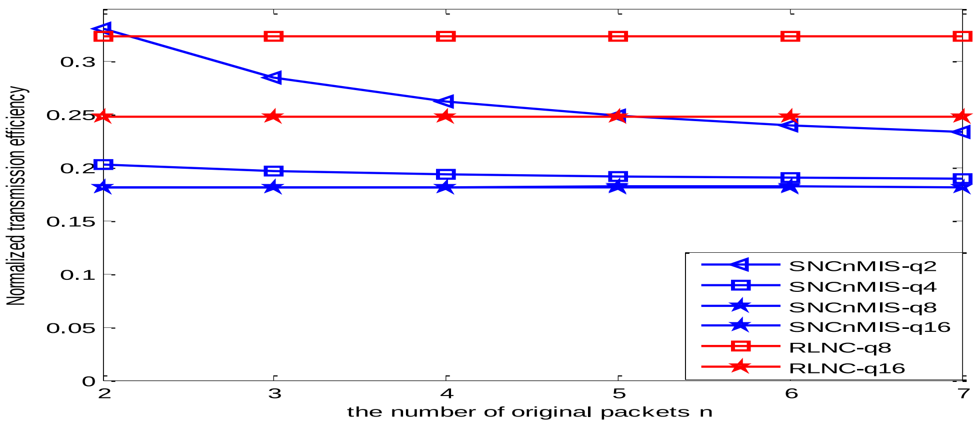

Fig. 2.The transmission efficiency of RLNC and SNCnMIS when L=50 bytes and σ=10.

In Fig. 2, the packets length L=50, then the $E(P)=0.1813$. The performances of SNCnMIS and RLNC are both become worse mainly because the packets loss ratio increases. Another reason just for SNCnMIS is that $C_{line}-k-n$ becomes small, a even negative number. Then SNCnMIS begins to lose effectiveness. σ has little effect.

The effectiveness of a sink receiving the last decoded original packets suffers two kinds of damages. One is the lost packets to counteract the channel noise and it is a unchanged price. Another is the loss of independency of coding vectors because coding coefficients are randomly selected. RLNC suffers the two both and the transmission efficiency is $q/(q-1)+k/n$. SNCnMIS makes the loss of in-dependency of coding vectors to be zero and its transmission efficiency is k/n if p_b is a constant.

5. Conclusion

This paper proposed a new distribution scheme based on so-called nMIS. It is reduces the computational complexity of encoding and decoding. With the least packets, this scheme increase the throughput , decrease the delay and consumed energy.

References

[1] J. Heide, M. V. Pedersen, F. H. Fitzek, and T. Larsen, "Network coding for mobile devices-systematic binary random rateless codes," in Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on, pp. 1-6, 2009.
[2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," Networking IEEE/ACM Transactions on, vol. 16, pp. 497-510, 2008.
[3] D. E. Lucani, M. Médard, and M. Stojanovic, "Random linear network coding for time division duplexing: Field size considerations," in in Proc. of IEEE GLOBECOM, pp. 1 - 6, 2009.
[4] G. Ma, Y. Xu, M. Lin, and Y. Xuan, "A Content Distribution System based on Sparse Linear Network Coding," Netcod, 2007.
[5] M. Luby, "LT codes," in Foundations of Computer Science, Proceedings, Annual IEEE Symposium on, pp. 271 - 280, 2002.
[6] L. I. Ling-Xiong, "Nonlinear Network Coding: A Case Study," Computer Science, vol. 2169, 2008.
[7] G. Zhang, "The generalization of Jaggi-Sanders algorithm to its nonlinear form," unpublished
[8] J. T. Seong, "Bounds on Decoding Failure Probability in Linear Network Coding Schemes with Erasure Channels," Communications Letters IEEE, vol. 18, pp. 648-651, 2014.
[9] D. E. Lucani, M. Stojanovic, and M. Me´Dard, "Random linear network coding for time division duplexing: When to stop talking and start listening," Proceedings IEEE Infocom, vol. abs/0809.2350, pp. 1800 - 1808, 2010.