Full length article

Performance evaluation of random linear network coding using a Vandermonde matrix

Pallavi R. Mane a, Sudhakara G. Adiga b, M. Sathish Kumar a, *

a Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal University, Manipal 576 104, India
b Department of Mathematics, Manipal Institute of Technology, Manipal University, Manipal 576 104, India

A R T I C L E   I N F O

Article history:
Received 11 September 2012
Accepted 4 November 2013
Available online 19 November 2013

Keywords:
Network coding
Utilization factor
Random linear network coding
Vandermonde matrix
Packet delivery delay

A B S T R A C T

This paper discusses random linear network coding with and without the use of a Vandermonde matrix to obtain the coding coefficients. Performance comparisons of such random linear network coded networks with networks employing traditional store and forward technique are also provided. It is shown that random linear network coding using a Vandermonde matrix can improve the network utilization factor by reducing the overhead compared to random linear coding that does not use a Vandermonde matrix. Our numerical results show that random linear network coding with a Vandermonde matrix provides a considerable improvement in throughput and delay when compared to a network employing a traditional store and forward strategy. An inherent feature of random linear network coding which makes it possible to employ simple encryption techniques is also discussed.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Network coding, in simple terms, is a technique that relies on combining independent packets at intermediate nodes and then forwarding them such that they could be recovered at their respective destinations. The technique is considered by several researchers as an innovative extension of the traditional store and forward paradigm and can effectively reduce packet density in communication networks. It increases the utilization of networks by reducing the expected number of transmissions and can be applied at symbol level in the physical layer, byte level in the MAC layer or at the packet level in a layer above the MAC layer. Network coding becomes robust in situations where the knowledge of topology is difficult to obtain or in the presence of link failures as in a Wireless Sensor Network (WSN) or in ad hoc networks. Due to its potential advantages, network coding has gained popularity in various wired and wireless applications and has been proposed for efficient data dissemination over WSN [1,2] and internet multimedia streaming [3,4].

A few other relevant and interesting studies on network coding are as follows. In [5], diversity network codes over finite fields based on linear network coding were applied for multiuser cooperative communication. For a network consisting of M users, it has been shown in [5] that a diversity of order $(2M - 1)$ can be achieved. Xiao et al. in [6,7] proposed a binary deterministic rate-less code for combination networks based on a sparse transfer matrix. Encoding is done using cyclic shifts and XOR’s. They show that the encoding complexity is linear. However, it can be used for only small set of packets.

Recently, Random Linear Network Coding (RLNC) [8,9] has received considerable attention from the research community as a technique to counter adverse effects as delays, fading, and erasures of wireless communication links. RLNC is attractive from the viewpoint of higher throughput and better utilization of network resources and is also one of the most suitable algorithms when network...
To outline briefly the idea behind RLNC, consider a source that has to transmit $M$ packets to a set of $N$ receivers through $N$ independent erasure channels as in Fig. 1. In RLNC, the encoder at the source linearly combines the $M$ packets and sends this combination as a single coded packet in the $j$th timeslot. Eq. (1) below represents the encoder operation for RLNC.

$$C_j = \sum_{i=1}^{M} \beta_{ij} P_i.$$  

(1)

Here $\beta_{ij}$ is the coding coefficient of the $i$th packet $P_i$, in the $j$th time slot. Usually, $\beta_{ij}$ is selected randomly from a Galois Field (GF) of appropriate size. At the receiver side, with the knowledge of the coefficients $\beta_{ij}$, the individual packets could be decoded from any $M$ coded packets received, provided the $M$ coefficients used to combine the packets to form these $M$ coded packets could be stacked to form an $M \times M$ invertible matrix. A major drawback here is that, since the coefficients are selected randomly, the invertibility condition mentioned above cannot be guaranteed. Secondly, the transmission of the $M$ coefficients $\beta_{ij}$ ($i = 1$ to $M$) to the receivers to facilitate recovery of packets adds to network overhead. Needless to say, both these drawbacks impact the throughput negatively. More details on RLNC over finite fields and its mathematical structure can be found in [10–12], and references therein.

To overcome such drawbacks of RLNC while retaining its advantages, an alternative approach to RLNC is proposed wherein the coded packet is generated with the coding coefficients as elements of the rows of a Vandermonde matrix [13–16]; we identify such RLNC which is based on Vandermonde matrix as RLNCV. It is well known that with the number of columns as $M$, any $M$ rows of a Vandermonde matrix will be linearly independent. This ensures the invertibility condition mentioned above.

RLNCV, like RLNC, is suitable for almost any network topology as it only differs in the way the coding vectors are generated and notably, generation of coding vectors is independent of network topology. An important significance of RLNCV considering a Vandermonde matrix for the generation of coding vectors is that there is no need to send all of the elements of the coding vector since an entire row of a Vandermonde matrix can be constructed using a single non-zero field element. This effectively compresses the network header to just one symbol as compared to $M$ symbols. It is worth mentioning here that to the best of our knowledge, no MDS code can compress the network header to such extent as achieved by using a Vandermonde matrix for generating the coding vectors.

Though usage of a Vandermonde matrix for network coding has been proposed earlier by other researchers, to the best of our knowledge, published literature does not have a detailed and comprehensive performance analysis of RLNC and RLNCV with appropriate comparisons to a traditional Store and Forward technique (SF). In this paper, we aim to bridge this important gap to a large extent. Our analysis and results quantify the performance of RLNCV as well as RLNC over finite fields. Our results also show rigorously the significant enhancements in network performance parameters such as throughput and delay, achieved by RLNC and RLNCV compared to SF. Further, we also show analytically the enhancement in network utilization factor achieved by RLNCV over RLNC.

In the remaining part of the paper, in Section 2 we compare RLNCV with RLNC with regards to the utilization factor and choice of coding coefficients. Later, in Section 3, we discuss some numerical results which bring out the performance of RLNCV and RLNC and comparisons of their performance with that of SF. Section 4 concludes the paper.

2. RLNCV and RLNC

Consider a system as earlier and as depicted in Fig. 1. With the traditional RLNC replaced by RLNCV, Eq. (1) which governs the packet encoding, gets altered to Eq. (2) as given below.

$$C_j = \sum_{i=1}^{M} (\beta_{ij})^{i-1} P_i.$$  

(2)
The coding coefficient $\beta_j$ should be non-zero and distinct for each time slot $j$ and can be selected from a GF of size $2^q$.

In the following analysis, we present a comparison of RLCNV and RLNC in terms of

1. Overhead and utilization factor.
2. Choice of coding coefficients.

**Overhead and utilization factor**

A typical structure of the coded packet for RLCNV is as shown in Fig. 2 (protocol header not depicted). Let the information to be shared be of size $I$ bits. This information is divided into $G$ generations with each generation having $M$ packets in them. Each packet is restricted to $L$ bits (a typical IP packet size excluding headers is $1400 \times 8$ bits). $I$ bits of information may or may not be multiples of $D$, where $D$ is as shown in Fig. 2., hence the number of packets in the last generation $(G+1$th generation) is $M_t \leq M$. If $S$ bits represent the number of packets in a generation, with the largest possible value of $S$ bits to be $M$ and $H$ the size of encoding coefficient overhead in each of the coded packet, the data size in a packet will be $D = (L - H - S)$.

For RLCNV the packet format as shown in Fig. 2, the utilization factor can be calculated as follows. Let $R$ be the bits of information remaining after forming the $G$ generations with each $M$ packets. Then

$$R = (I - GM_D)$$

with

$$G = \left\lfloor \frac{I}{MD} \right\rfloor$$

Tailing generation size $M_t$ can now be shown to be

$$M_t = \left\lfloor \frac{R}{D} \right\rfloor$$

and the total overhead $V_{RLNC}$ would be

$$V_{RLNC} = (H + S)(GM + M_t).$$

The utilization factor for RLCNV can now be readily shown as

$$U_{RLNC} = \frac{I}{V_{RLNC} + 1}. \quad (7)$$

On the other hand, for traditional RLNC with the structure of coded packet as given in [17], we can arrive at the RLNC utilization factor as follows.

As $L$, the length of the packet is fixed by the protocols used and $H$ increases with generation size, the choice of $D$ depends on the network overhead $H$. When $R < MD$, for the packets in the last generation, the number of data bits $D$ could be less, say $D_t$.

Thus, tailing generation size $M_t$ would be

$$M_t = \left\lfloor \frac{R}{D_t} \right\rfloor$$

and the total overhead $V_{RLNC}$ would be

$$V_{RLNC} = (GM^2 + M^2 H) + S(G + 1). \quad (9)$$

The utilization factor for RLNC can now be readily shown as

$$U_{RLNC} = \frac{I}{V_{RLNC} + 1}. \quad (10)$$

**Choice of coding coefficients**

In RLNC, the coding vector, which comprises the coefficients $\beta_j$ and its exponents, is obtained from the row of a Vandermonde matrix. Hence, it is obvious that it is sufficient to choose only one coefficient ($\beta_j$) for a particular time slot. It has to be ensured that this coefficient is not repeated in subsequent time slots. If the coefficient is chosen from GF ($2^q$), number of choices of coding vectors would be $2^q - 1$. Obviously, the all zero vector has to be avoided.

In RLNC, on the other hand, the receiver can recover the $M$ packets from coded packets if the matrix formed by horizontally stacking the coding vectors is of rank $M$. Staying within this inevitable constraint and exploring the options, it has been shown in the Appendix that the maximum number of coding vectors available is not more than $1 + M + M(M - 1) \left\lfloor \frac{2^q - 1}{2} \right\rfloor$, such that any $M$ vectors selected from this set of $1 + M + M(M - 1) \left\lfloor \frac{2^q - 1}{2} \right\rfloor$ coding vectors will be linearly independent.

### 3. Results and discussions

For a typical value of $I = 500$ MB and with an error free channel, let the packet size be fixed as $1400$ B. Now using coding coefficients from GF ($2^q$) and using the relations developed in Eqs. (3)–(10), it can be shown that for RLCNV the utilization factor $U_{RLNC}$ is 99.93% and is independent of generation size, while for the RLNC, the utilization factor $U_{RLNC}$ is 96.42% and 81.77% for generation sizes of 50 and 255 respectively.

To evaluate the performance in terms of the throughput and delay, the system is supposed to have an ARQ strategy with an ACK that conveys to the source the degrees of freedom (dof) [17–20]. To model the effect of erasure, we generate an erasure matrix similar to the one in [21]. An example is shown in Table 1. The finite field considered is GF ($2^8$) and we use the brute force technique discussed in [22].

Fig. 3 depicts the RLCNV algorithm in the context of the scheme under consideration and as shown in Fig. 1. The fundamental idea is that a new coded packet is transmitted in each time slot until an ACK ‘dof = 0’ is not received from all the receivers. Thus, the process is completed when all the packets are decoded successfully.
Table 1
Erasure matrix.

<table>
<thead>
<tr>
<th>Time slots</th>
<th>(j)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>–</th>
<th>–</th>
<th>j ≥ M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rx 1</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Rx 2</td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Rx 3</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>–</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rx (N − 1)</td>
<td></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Rx N</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig. 3. The RLNCV algorithm.

To demonstrate the numerical results, we first consider a network with 1 Source and 6 Receivers (1S6R). Figs. 4–6 show the throughput, packet delivery delay and overall delay (decoding delay for all the M packets of one generation) respectively, as a function of generation size. Results are shown for networks that use RLNCV as well as networks that use SF. It can be observed from these results that up to a generation size of around 6, RLNCV as well as traditional store and forward scheme perform almost the same. However, above a generation size of 6, RLNCV starts to show a significant improvement in performance.

Figs. 7–9 depicts the same parameters as in Figs. 4–6. However, the parameters here are plotted as a function of number of receivers with the generation size fixed as 12 and the number of sources to one. Here as well, it is seen that the RLNCV considerably outperforms the SF.

It may be specifically noted that as far as the parameters plotted in the graphs shown in Figs. 4–9 are concerned, both RLNC and RLNCV’s results hold good for RLNC as
that RLNC is better than RLNC when it comes to network utilization factor as shown in the beginning of this section.

Before we conclude, we would like to highlight some additional interesting features of RLNC as well as RLNCV. Since RLNCV needs to send only a single element in order to facilitate the receiver to decode the packets, the decoding delay in RLNCV may be less compared to that in RLNC. Secondly, both RLNC as well as RLNCV have an inherent security feature in them. This could be appreciated if one considers not sending the coding coefficients (all the $M$ coefficients in RLNC or just the one coefficient in RLNCV) as it is but do some simple encryption on them prior to sending them. For example, in RLNCV, a possible technique could be to transmit the multiplicative inverse or additive inverse of the coding coefficient in a pseudo random manner in each of the packets.

4. Conclusions

In this paper we have provided a comparison between RLNC and RLNCV (RLNC with Vandermonde). The comparisons have clearly highlighted the advantages of RLNCV over RLNC in terms of a reduced overhead and consequent improvement in network utilization factor. Our analysis and numerical results have addressed important gaps that exist in published literature in the form of a lack of a comprehensive comparison between RLNC, RLNCV and traditional SF technique. Numerical results presented in this paper show the improvements in network performance parameters such as throughput, packet delivery delay and the overall delay brought about by RLNCV compared to the traditional SF technique. Certain other beneficial features offered by RLNC and RLNCV over SF such as the possibility of simple encryption techniques have been briefly discussed. Further investigations on these features in particular and on other relevant aspects of RLNCV will be useful considering the overall potential of RLNCV and the ever increasing interest in network coding.

Appendix

In RLNC, the source selects $M$ random coefficients to generate the encoded packet in every time slot. The cardinality of the set of vectors with $M$ elements is $(2^q - 1)^M$. Let $V$ be the subset of this set of vectors such that any $M$ vectors chosen from this subset are linearly independent. In the following, we develop certain important constraints on the coding matrix such that it is invertible, and try to find the size of the subset $V$.

1. Consider a coefficient vector $C_i = (\beta_{i1}, \beta_{i2}, \ldots, \beta_{iM})$ from the subset $V$ which is from the subset $V$ which is chosen for $i$th time slot with $\beta_{iv}$ a non-zero element from $GF(2^q)$ and belonging to $V$. Then for the $j$th time slot we cannot choose a coding vector which is of the form $C_j = aC_i$, as it does not enhance the information required for decoding at the receiver.

2. Any $M$ vectors chosen from $V$ should not have two or more columns equal as the matrix of coding vectors will be singular and hence non-invertible.
(3) Any $M$ vectors chosen from $V$ should be linearly independent.

(i) Say initially the coding vector chosen has all the equal coefficients. For the subsequent time slot the coding vector cannot have all coefficients equal due to constraint 1.

(ii) Coding vector having all the coefficients equal except one. The vectors resemble one of the rows of the matrix given below. Only $M$ vectors of this type can be selected at random such that it satisfies all the three constraints.

\[
\begin{pmatrix}
bbb
b
b
b
b
\end{pmatrix}
\]

where $a \neq b; a, b \in GF(2^q)$.

(iii) Coding vectors having $(M - i)$ equal coefficients and $i$ distinct coefficients, where $i = 1$ to $(M - 2)$ resemble the rows of the matrix given below. Only $M$ vectors of this type can be selected at random such that they satisfy all the three constraints.

\[
\begin{pmatrix}
bbb
b
b
b
b
\end{pmatrix}
\]

has rank $M$.

\[
\begin{pmatrix}
c
a
b
\end{pmatrix}
\]

$\beta_{i1} = b, \beta_{i2} = c, \ldots, \beta_{iM} = a$;

where $a, b \in S$ and $a \neq b$.

(iv) When coding vectors have all distinct coefficients, they take the form of rows in the matrices given below

For $a \neq b \neq c \neq d$; $a, b, c, d \in GF(2^q)$

\[
\begin{pmatrix}
a
b
\end{pmatrix}
\]

or

\[
\begin{pmatrix}
a
b
\end{pmatrix}
\]

and so on. All have full rank $M$.

Thus there can be either 1 vector from (i), $M$ vectors from either (ii) or (iii), and $M(M - 1)\binom{2^q-1}{2}$ from (iv). Thus, $V$ can have a maximum of $1 + M + M(M - 1)\binom{2^q-1}{2}$ vectors.

References

M. Sathish Kumar graduated in Electronics and Communication Engineering from the Institute of Engineers (India) in the year 1992 and obtained his Masters from the College of Engineering, Guindy, Anna University, in the year 1996 specializing in Optical Communication. He earned his Ph.D. in Electronics and Communication Engineering from the National Institute of Technology, Karnataka in 2004 for his thesis on the performance enhancement of optical CDMA networks and hybrid multiplexed optical communications systems. He has about 15 years of academic experience in different capacities in the Dr. T.M.A. Pai Polytechnic, Manipal Institute of Technology, Manipal University and the School of Electrical Engineering and Computer Science, Seoul National University. Currently he is a professor in the Electronics and Communication Engineering Department of Manipal Institute of Technology, Manipal University.

He was the recipient of the International Scholar Exchange Fellowship of the Korea Foundation for Advanced Studies for the year 2006–07. His areas of interests are advanced modulation formats for optical communications, optical CDMA systems, modelling and optimization of nanophotonics devices, and general areas of digital communications. He has authored or co-authored around 50 papers in referred journals and conference proceedings.