AFRL-RI-RS-TR-2013-247

# WIRELESS COMPUTATIONAL NETWORKING ARCHITECTURES

*DECEMBER 2013*

FINAL TECHNICAL REPORT

***APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED***

STINFO COPY

## AIR FORCE RESEARCH LABORATORY
## INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

# NOTICE AND SIGNATURE PAGE



AFRL-RI-RS-TR-2013-247 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

**/ S /**

JOSEPH A. CAROLI, Chief
High Performance Systems Branch

**/ S /**

MARK H. LINDERMAN
Technical Advisor
Computing & Communications
Information Directorate

# REPORT DOCUMENTATION PAGE

*Form Approved*
**OMB No. 0704-0188**



**1. REPORT DATE** *(DD-MM-YYYY)*
DECEMBER 2013

**2. REPORT TYPE**
FINAL TECHNICAL REPORT

**3. DATES COVERED** *(From - To)*
JAN 2007 – JUL 2013

**4. TITLE AND SUBTITLE**
WIRELESS COMPUTATIONAL NETWORKING ARCHITECTURES

**5a. CONTRACT NUMBER**
IN-HOUSE

**5b. GRANT NUMBER**
N/A

**5c. PROGRAM ELEMENT NUMBER**

**5d. PROJECT NUMBER**
WCNA

**5e. TASK NUMBER**
00

**5f. WORK UNIT NUMBER**
01

**6. AUTHOR(S)**
Bruce W. Suter

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Air Force Research Laboratory/RITB
525 Brooks Road
Rome NY 13441-4505

**8. PERFORMING ORGANIZATION REPORT NUMBER**
N/A

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Air Force Research Laboratory/RITB
525 Brooks Road
Rome NY 13441-4505

**10. SPONSOR/MONITOR'S ACRONYM(S)**
AFRL/RI

**11. SPONSORING/MONITORING AGENCY REPORT NUMBER**
AFRL-RI-RS-TR-2013-247

**12. DISTRIBUTION AVAILABILITY STATEMENT**
Approved for Public Release; Distribution Unlimited. PA# 88ABW-2013-5383
Date Cleared: 18 Dec 13

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**
Motivated by the need of providing on-demand local computing resources over a wireless computing and communication infrastructure, we will begin to face over the next decade very large sensor-generated datasets on the order of zettabytes or even yottabytes. While centralized control and applications may still be needed, the bulk of the processing and storage must be distributed near the sensors. Motivated by results in compressive sampling, we obtain new results in rank deficient network coding, which is capable of addressing some of these challenges.

**15. SUBJECT TERMS**
Wireless Network Computations, Wireless Network Architectures, Distributed Computing, Sensor Networks

| 16. SECURITY CLASSIFICATION OF: a. REPORT | b. ABSTRACT | c. THIS PAGE | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| U | U | U | UU | 20 | **BRUCE W. SUTER** |
| | | | | | **19b. TELEPONE NUMBER (*Include area code*)** N/A |

# TABLE OF CONTENTS

**Section** **Page**

## LIST OF FIGURES

**Figure** **Page**

# 1.0 SUMMARY

Motivated by the need of providing on-demand local computing resources over a wireless computing and communication infrastructure, we will begin to face over the next decade very large sensor-generated datasets on the order of zettabytes, i.e. $10^{21}$ bytes, or even yottabytes, i.e. $10^{24}$ bytes. While centralized control and applications may still be needed, the bulk of the processing and storage must be distributed near the sensors. Motivated by results in compressive sampling, we obtain new results in rank deficient network coding, which is capable of addressing some of these challenges. Although rank deficient network coding is one of the most significant results of the in-house project, many other results were obtained. Highlights of the WCNA project are summarized below.

## 1.1 Compressive Sampling Research

We considered the problem of compactly representing big data. Initially, we looked at novel ways to improve encoders and decoders for “classical compressive sampling.” More recently, we have focused on 1-bit compressive sampling. We chose the sign bit as a basis since it represents minimal storage and it would be insensitive to dynamic range. Previously known approaches required sparsity to be known a priori, which is not always practical. However, prior knowledge of signal sparsity is not required by the 1-bit novel approach to compressive sensing; thus, allowing blind 1-bit compressive sensing - to be a practical system-level paradigm for big data.

## 1.2 Synchronization Research

We modeled the synchronization process between sensors within a sensor network by addressing the “clock slew” problem. The resulting solution significantly increases the coherency interval after re-synchronization while making current standard protocols more efficient. Moreover, the cost in power consumption to implement is very small while actually reducing the overall power consumption of the sensor network; thus extending the sensor’s usefulness.

## 1.3 Network Coding Research

Recently, We have taken on the challenge of efficient distribution of big data to multiple locations. This emphasis uses linear combinations of data packets allowing the distribution process to be insensitive to the receipt of a particular data packet. Some research has involved the development of a Very Large Scale Integration (VLSI) realization of network coding decoders with error correction. More recently, we have studied previous network coding approaches that require a full rank coefficient matrix of linear combinations before decoding could begin; thus, leading to long delays and low throughput. We developed and submitted a patent on a revolutionary approach for a rank deficient decoding scheme that takes advantage of the sparsity inherent in data to estimate data packets. This process makes rank deficient decoding of network coding a useful system-level paradigm for big data. The rest of this report focuses on this notable achievement.

## 2.0 INTRODUCTION

This project had the goal of providing on-demand computing resources over wireless computing, communication and sensing infrastructures. We aimed to develop new system-level paradigms called Rank Deficient Network Coding that is a novel approach to network coding inspired by compressive sampling. The rest of this section provides an introduction to the issues surrounding rank deficient network coding.

Network coding is a new paradigm of communication networks that promises advantages in throughput, robustness, and complexity. Since the fundamental premise of linear network coding is that transmitted data packets are subject to linear combinations, for all previous network coding schemes, a full rank of received packets is required to invert the linear mapping so as to recover the transmitted data packets. This requirement unfortunately results in a key drawback of "classical" network coding: either all the packets (or bits) in a session are recovered simultaneously or none can be recovered. Aiming to overcome this all-or-nothing property, which has long delays and low throughput, in this report we present a variety of rank deficient decoders of linear network coding. To this end, we reformulate the decoding problem of linear network coding as a collection of underdetermined systems. This reformulation reveals the connection among the decoding problems of network coding error control coding, and enables rank deficient coding. We then present two classes of rank deficient decoders. The first class of decoders takes advantage of the sparsity inherent in the data and produces the data vectors with the smallest Hamming weight, and hence they are called Hamming norm decoders. These decoders have a high complexity, so we present a class of decoders based on linear programming, referred to as linear programming decoders. Considering linear programming relaxation of the Hamming norm decoders and solving them using standard linear programming procedures, the linear programming decoders have polynomial complexities and are much more affordable. Both classes of decoders recover data from fewer received packets and hence achieve higher throughputs and shorter delays than the full rank decoder.

Communication networks (CNs) are ubiquitous in our everyday life as well as our national infrastructure. Network coding [1] has the potential to fundamentally transform current and future CNs due to its promise of significant throughput gains. Furthermore, network coding has other advantages such as robustness and can be implemented in a distributed manner with random linear network coding (RLNC) [2]. Hence, network coding is already used or considered for a wide variety of wired and wireless networks.

Although network coding does not suffer from the coupon-collector problem, one significant drawback of network coding is its all-or-nothing property in more than one sense. First, a full rank of received packets at the receiver nodes of a multicast (or a unicast) is needed before decoding can start, leading to long delays and low throughputs, especially when the number of packets of a session is large. This is particularly undesirable for military applications with stringent delay requirements. Second, all the bits in any packet are equal in the sense that they are recovered simultaneously.

## 3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

In this section, we describe assumptions and methodologies we used in the development of the rank deficient decoder for linear network coding.

Aiming to solve the problems of "classical" network coding, we developed rank deficient decoding for linear network coding, which can start even when the rank of the received packets is smaller than the full rank. By reformulating the decoding problem of network coding in a different fashion, the decoding problem reduces to a collection of syndrome decoding problems, rank deficient decoding leads to smaller delays and higher throughput, at the expense of possible decoding errors.

We developed two classes of rank deficient decoders. The first class of decoders takes advantage of the sparsity inherent in the data and produces the data vectors with the smallest Hamming weight, and hence they are called Hamming norm decoders. These decoders have high complexities, so they are not practical for large systems. The second class of decoders developed was based on linear programming and, as such, they will be referred to as linear programming decoders. Considering linear programming relaxation of Hamming norm decoders and solving them using standard linear programming procedures, the linear programming decoders have polynomial complexity and, as such, are much more affordable. Both classes of decoders recover data from fewer received packets and hence achieve higher throughput and shorter delays than the "classical" full rank decoder. Since these decoders could produce erroneous outputs, within each class different strategies can be proposed with different tradeoffs between delay/throughput and data accuracy, and they include the traditional decoder of network coding as a special case.

In the literature, there are two different approaches to deal with the synergy of network coding and compressive sensing, and they aim for different applications. One approach proposed [3] uses the statistical property of data blocks to alleviate the "all-or-nothing" drawback of network coding in distributed storage systems. In this approach, random linear network coding is carried out over finite fields and the data is represented in bits. The other approach [4] [5] aims to take advantage of the statistical correlation of data generated by distributed sensor networks. In this approach, data are real values and linear combinations are taken over real (or complex) fields. The rationale is that the real representation of data is more natural for sensor networks [4] [5]. In practice, data are represented in a finite precision system. It has been shown that information loss due to finite precision grows with the network size [6].

Our research at AFRL is significantly different from both existing approaches. Above all, our reformulation of the decoding problem is novel, and this reformulation was not previously considered. Furthermore, [3] focuses on the application of random linear network coding in distributed storage systems. In contrast, we consider linear network coding in general, and our work applies to a wide variety of applications. Also, network coding is classically carried out over the real (or complex) field, for example [4] [5]; whereas in our work, network coding is carried out over finite fields. Thus, our scheme does not suffer the information loss due to finite precision as in [4] [5].

## 3.1 Systems Model

In this work, we make several assumptions about the underlying network coding. First, we treat all packets as N-dimensional row vectors over some finite field GF(q), where q is a prime  For simplicity, as most network coding schemes in practice, we assume that GF(q) is a finite field of characteristic two, because information (in bits) can be easily mapped to finite field symbols. Second, we focus on linear network coding (LNC) [7] only, which was shown to be optimal in most cases. Thirdly, we assume that the network is error-free, and error control (see, for example, [8]-[11]) is not embedded in network coding.

Suppose a source node of a unicast or multicast injects a collection of *n* data packets (or vectors over GF(q), $X_0, X_1 ..., X_{n-1}$, into the network. At any sink node, *m* packets (or vectors over GF(q)), $Y_0, Y_1 ..., Y_{m-1}$, are received, where $Y_i = \sum_{j=0}^{n-1} a_{i,j} X_j$ for $i = 0, ..., m-1$ and $a_{i,j} \in GF(q)$. In other words, each received packet is a linear combination of injected packets. Since the sink node can locally generate more linear combinations of $Y_0, Y_1 ..., Y_{m-1}$, it is assumed that $Y_0, Y_1 ..., Y_{m-1}$, are linearly independent, which implies that $m \le n$. That is, the *m* x *n* matrix *A*, where $a_{i,j} \in A$, has a rank of *m*. The matrix *A* is sometimes called the global coding kernel matrix.

## 3.2 Full Rank Decoding

Let us further denote the matrices $\begin{bmatrix} X_0^T & X_1^T & \cdots & X_{n-1}^T \end{bmatrix}^T$ and $\begin{bmatrix} Y_0^T & Y_1^T & \cdots & Y_{m-1}^T \end{bmatrix}^T$ as *X* and *Y*, respectively, and they are related by $Y = AX$. The sink node can recover the transmitted data packets by reversing the encoding of the data packets by the network. This is achievable when $m = n$, as the sink node can recover the data packets by computing $X = A^{-1}Y$. Thus, the decoding in network coding starts only after the sink node has received enough linearly independent combinations of the transmitted data packets. That is, either all the data packets are recovered simultaneously, or none are recovered. This is often referred to as the “all-or-nothing” property in network coding. Note that this is different from the coupon collector problem suffered by communications networks without network coding. Nevertheless, the required number of linearly independent packets received by the sink node leads to longer delays and lower throughputs, which may be undesirable for some applications. Furthermore, in this setting, the "all-or-nothing" property also holds on the bit level. That is, all bits of all packets are equal in the sense that either all of them are recovered simultaneously or none of them can be recovered.

## 3.3 Rank Deficient Decoding

We can formulate the data recovery problem at the sink node in a different way. Let us consider coordinate $k$ of $Y_i$, and we have $Y_{i,k} = \sum_{j=0}^{n-1} a_{i,j} X_{j,k}$ for $i = 0,1,\cdots,m-1$ and $k = 0,1,\cdots,N-1$. Let us denote the column vectors $\left(Y_{0,k} \quad Y_{1,k} \quad \cdots \quad Y_{m-1,k}\right)^T$ and $\left(X_{0,k} \quad X_{1,k} \quad \cdots \quad X_{n-1,k}\right)^T$ as $Z_k$ and $W_k$, respectively. Clearly, we have $Z_k = AW_k$ for $k = 0,1,\cdots,N-1$. The sink node can recover the data packets if it can obtain $W_k$ from

$$Z_k = AW_k \text{ for } k = 0,1,\cdots,N-1. \tag{1}$$

Eq. (1) shows that the data recovery problem at the sink node can be viewed as N parallel decoding problems, and each corresponds to one coordinate in the packet (or vector). When these N parallel decoding problems are solved at the same time, it is essentially equivalent to the traditional decoding problem of network coding.

This reformulated problem is related to two well-known decoding problems. First, if we treat the *m* x *n* matrix *A* as a parity check matrix for a linear block code of length *n* and dimension *n-m*, the decoding problem in Eq. (1) is closely related to a syndrome decoding problem. That is, the sink node needs to recover for $W_k$ based on the syndrome $Z_k$. Second, if we treat $W_k$ as a data vector and *A* as a measurement matrix, this is analogous to the decoding problem in compressive sampling.

## 3.4 Decoding Strategy

Once a full rank of received packets is available, the full-rank decoder recovers all data packets correctly. In contrast, the rank deficient decoders may produce wrong decisions. Analogous to classical error control coding, the preference between decoding failures and decoding errors varies from one application to another. For instance, for military applications with stringent delay constraints, partially correct data packets may be more desirable than decoding failures. For other applications, such as cloud storage, data integrity may be more important than delays, especially if packet retransmission is possible. Hence, it is necessary to consider a wide range of decoding strategies so as to offer different tradeoffs between delays/throughput and accuracy.

Two extreme strategies are natural and straightforward. One extreme, called the error-free decoder, is similar to the full rank decoder in the sense that it decodes only if decoding success is guaranteed. The other extreme, referred to as the best-effort decoder, always tries to decode with available received packets. The error-free and best-effort decoders represent the most conservative and most aggressive strategies.

### 3.5 Hamming Norm Decoders

Let us further consider the problem in Eq. (1). Since the data recovery problem at any sink node is equivalent to a collection of parallel problems in Eq. (1), we focus on one such problem. In other words, we try to solve $Z = AW$ for $W$, where $Z$ and $W$ are $m$-dimensional and $n$-dimensional column vectors, respectively, and $A$ remains an $m$ x $n$ matrix with full rank. Without loss of generality, we assume that $m < n$.

For a linear block code of length $n$ and dimension $n$-$m$ with a parity check matrix, $A$, then $Z = AW$ can be viewed as a syndrome of the received vector $W$. It is well-known that for a linear block code, every vector in a coset has the same syndrome and different cosets have different syndromes. Thus, solving $Z = AW$ for $W$ is equivalent to finding a vector within a coset.

If no side information is available, we can make a decision within the coset by taking advantage of some inherent properties of the data vector. In this work, we proceed by relying on the sparsity of the data vector, which is well justified in many applications. That is, the proposed rank deficient decoders produce the vector with the smallest Hamming weight in the coset. Hence, we refer to them as Hamming norm decoders.

As is common in the compressive sampling literature, we consider two possible scenarios for sparsity. First, when $W$ is sparse, we use a vector with the smallest Hamming weight in the coset corresponding to $Z$ as an estimate of $W$. Second, suppose that $\Phi W$ is sparse for a known nonsingular $n$ x $n$ matrix $\Phi$. Since $Z = AW = A\Phi^{-1}\Phi W$, we can treat Z as a syndrome for the linear block code defined by $A\Phi^{-1}$. Thus, in this scenario, we first select a vector with the smallest Hamming weight in the coset corresponding to Z, and then produce an estimate of W by multiplying the selected vector with $\Phi^{-1}$. In both scenarios, the key step is to select a vector with the smallest Hamming weight in the coset corresponding to the given syndrome. Thus, we assume $W$ is sparse without loss of generality.

In coding theory terminology, a vector with the smallest Hamming weight among a coset is called a leader of the coset. Note that some coset leaders may not be unique, when more than one vector in the coset has the smallest Hamming weight. In this case, either the coset leader is selected among these vectors at random or a list of all potential leaders.

We remark that this problem is closely related to but different from the syndrome decoding problem in classical coding theory. In our decoding, a vector or a list of vectors with the smallest Hamming weight in the coset corresponding to the given syndrome is considered as the estimate of the data vector. In the syndrome decoding problem, a coset leader is often considered as an estimate of the error vector. However, the key step in both problems is to select a vector or a list of vectors with the smallest Hamming weight in the coset corresponding to the given syndrome. For this reason, we refer to our decoding problem as the modified syndrome decoding problem.

Thus, we have the following sufficient condition for successful decoding:

**Lemma 1.** The minimum Hamming distance of the linear block code defined by *A*, denoted by $d_H(A)$, satisfies $d_H(A) \le m+1$. When the Hamming weight of *W*, denoted by $w_H(W)$, is less than half of the minimum Hamming distance of the linear block code defined by *A*, that is $w_H(W) < \frac{d_H(A)}{2}$, *W* can be recovered by syndrome decoding.

*Proof:* The first part is due to the Singleton bound on the minimum Hamming distance of linear block codes. The second part holds because it is well known that a coset leader with Hamming weight less than $\frac{d_H(A)}{2}$ is unique.

Q.E.D.

When *W* is not a unique coset leader, there are two possibilities. First, when the Hamming weight of *W* is minimal in its coset, either *W* has a probability to be selected when coset leaders are chosen at random or *W* is one of the possible vectors produced by the decoder, depending on whether the decoder needs to generate only one vector or a list of vectors. Second, when the Hamming weight of *W* is not minimal, a wrong vector will be produced by the modified syndrome decoder.

## 3.6 Linear Programming Decoders

Since both the computational complexity and the memory requirements of the Hamming norm decoders grow exponentially with the size of *A*, we also adopt a linear programming (LP) approach. Since *A* is not necessarily sparse, we formulate the problem based on that for binary linear block code with high-dense polytopes [12].

An *m* x *n* parity-check matrix *A* can be represented by a Tanner graph *G*, a bipartite graph with a set of variable nodes $I = \{1, 2, \ldots, n\}$ and a set of check nodes $J = \{1, 2, \ldots, m\}$. A node $i \in I$ is adjacent to a node $j \in J$ if the element $a_{i,j} \in A$ is nonzero. *N(j)* is the set of variable nodes that are adjacent to check node *j*, and *N(i)* is the set of check nodes adjacent to variable node *i*.

Let $f_1, f_2, \ldots, f_n$ be the variables representing the code bits of w, and $s = \left(s_1, \quad s_2, \quad \ldots, \quad s_m\right)^T$ be the syndrome received. For each check node $j \in J$, let $T_j^E = \{0, 2, 4, \ldots, 2\lfloor |N(j)|/2 \rfloor\}$ for $s_j = 0$, and $T_j^O = \{1, 3, 5, \ldots, 2\lfloor (|N(j)|-1)/2 \rfloor + 1\}$ for $s_j = 1$. Then, for each $j \in J, k \in T_j^E (T_j^O)$, and $i \in N(j)$, define $z_{i,j,k}$. Then the linear programming formulation for the syndrome decoding is to minimize $\sum_{i=1}^{n} f_i$, subject to the following constraints:

$$\forall i \in I, j \in N(i), \quad f_i = \sum_{k \in T_j} z_{i,j,k}$$

$$\forall j \in J, \quad \sum_{i \in N(j)} z_{i,j,k} = k\alpha_{j,k}$$

$$\forall i \in I, \quad 0 \le f_i \le 1$$

$$\forall j \in J, k \in T_j \quad 0 \le \alpha_{j,k} \le 1$$

$$\forall i \in I, j \in N(i), k \in T_j, \quad 0 \le z_{i,j,k} \le \alpha_{j,k}.$$

The above constraints are similar to those in [12], except that $T_j = T_j^E$ in the previous constraints if $s_j = 0$, and $T_j = T_j^O$ if $s_j = 1$. In addition, the following constraint is added to narrow down the optimal solutions:

$$\sum_{i=1}^{n} f_i \ge cw$$

Linear programming (LP) may produce non-integral results, in which two approaches are considered. The first type is simply to round off the real numbers into integers, which are compared with the original data to count decoding error or success rate, and we mark this approach LP I. The other one (LP II) is to declare decoding failure of the entire generation, as the decoding is performed column wisely, and each packet in the same generation will be affected. Both LP I and LP II are applicable to all greedy as well as the error free (EF) and best effort (BE) strategies.

## 3.7 Implementation Considerations

The received packets are first processed to remove any linearly dependent packets. Then, the decoding problem is decomposed into multiple sub-problems. After solving the sub-problems in parallel, the solutions to the sub-problems are used to reconstruct the transmitted data packets.

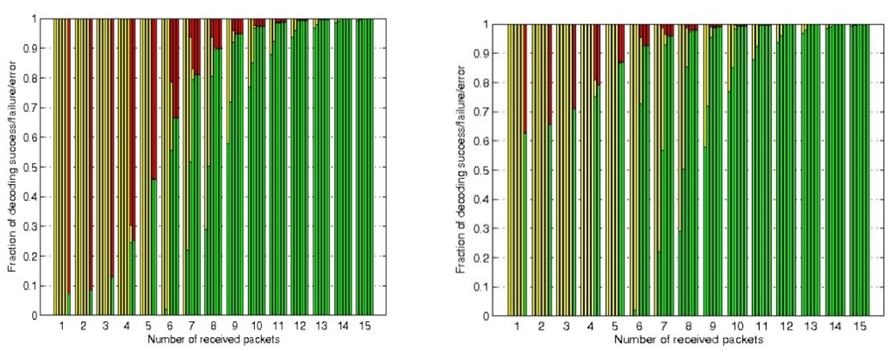
Each received packet has two parts relevant to linear network coding: the first, called the header, contains the linear coefficients used to obtain the received packet, and the second is the data. Note that the data is the linear combination of the transmitted data packets using the coefficients in the header.

The decoding problem is now reformulated into N parallel sub-problems: $AW_i = V_i$ for $i = 0, 1, \cdots, N-1$, where $V_i$ is the $i$-th column of $V$. Then, we identify a set of possible solutions using a property of the data to pick the solution. For instance, if it is known that the data packets are sparse, then the solution(s) with a certain number of nonzero entries will be selected. After all N sub-problems are solved; the solutions to all sub-problems are collected**.** Then, we determine whether every sub-problem has a single solution. If so, then $W_i$'s have been obtained from the sub-problems, and they are collected to form the transmitted data packets. If multiple candidates exist, they form the set of possible data packets, and then select the transmitted data packets using additional data packets.

## 4.0 RESULTS AND DISCUSSIONS

In our early simulations, $n = 8$ transmitted packets of length $N = 8$ are generated such that the transmission matrix has a constant column weight of $cw = 2.$ Note that such small parameters are chosen so that the complexities of the Hamming norm decoders are manageable. The matrix $A$ is generated randomly, with each element being 0 or 1 with equal probability. The number of received packets m varies from 1 to 15, while the packets may not be linearly independent. Simulation results are obtained based on 100,000 generations of packets injected into the network.

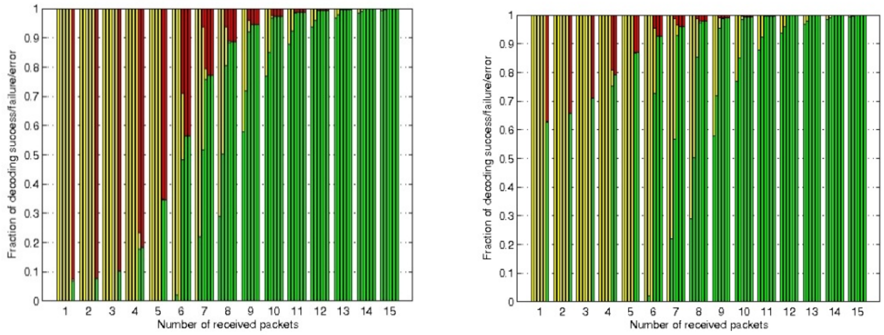
Fig. 1(a) and Fig. 1(b) show the packet level and bit level performance of different decoding algorithms, where the syndrome decoding adopts the hamming norm decoding algorithm. Both the packet success rate (PSR) and bit success rate (BSR) approach 1 following increased number of received packets for the syndrome decoding and the LNC decoding. But LNC performs no decoding when the number of received packets m is smaller than the number of transmitted packets $n$. Further, when $m \geq n,$ the syndrome decoding algorithm achieves much better results than the traditional LNC for both the packet and bit levels performance.




(a) (b)

Figure 1. Hamming Norm Decoding (n=N=8) at the (a) packet level and (b) bit level

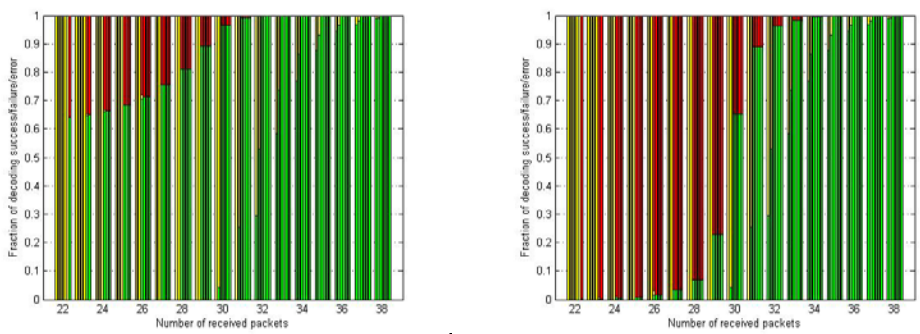
Simulation results obtained from the linear programming decoding algorithm for syndrome decoding are shown in Fug. 2(a) and Fig. 2(b). As expected, the linear programming approach performs slightly worse compared to the hamming norm decoding algorithm. However, the performance difference vanishes when the number of received packets is large enough.




(a) (b)

Figure 2. Linear Programming Decoding (n=N=8) at the (a) packet level and (b) bit level

Fig. 3 illustrates the scalability of the linear programming based approach. For Fig. 3, $n = 32$ transmitted packets of length $N = 32.$ Although the number of packets has increased by a factor of four, Fig. 3 has the same overall behavior as Fig. 2.




(a) (b)

Figure 3. Linear Programming Decoding (n=N=32) at the (a) packet level and (b) bit level

## 5.0 CONCLUSIONS

We have presented a variety of rank deficient decoders of linear network coding. Compared with the full rank decoder universally used in linear network coding, our proposed decoders require fewer received packets to decode and hence to achieve higher throughput and shorter delays. Hence, these new results on rank deficient network coding begin to address the challenges of *big data*.

## 6.0 REFERENCES


[1] R. Ashlswede, N. Cai, S. Li, and R. Yeung, "Network Information Flow," IEEE Transactions on Information Theory, Vol. 46, pp. 1204-1216, July 2000.

[2] T. Ho, M. Medard, R. Kotter, D. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," IEEE Transactions on Information Theory, Vol. 5, pp. 4413-4430, October 2006.

[3] H. Chen, "Distributed File Sharing: Network Coding meets Compressed Sensing," Proceedings of First International Conference on Communications and Networking in China, pp. 1-5, Beijing, China, October 2006.

[4] S. Katti, S. Shintre, S. Jaggi, D. Katabi, and M. Medard, "Real Network Coding," Proceedings of Forty-Fifth Annual Allerton Conference in Communications, Control, and Computing, pp. 389-395, Allerton House, Illinois, September 2007.

[5] N. Nguyen, D. Jones, and S. Krishnamurthy, "Netcompress: Coupling Network Coding and Compressed Sensing for Efficient Data Communication in Wireless Sensor Networks," Proceedings of 2010 IEEE Workshop on Signal Processing Systems (SiPS 2010), pp. 356-361, San Francisco, California, October 2010.

[6]S. Shintre, S. Katti, S. Jaggi, B. K. Dey, D. Katabi, and M. Medard, "Real and Complex Network Codes: Promises and Challenges," Fourth Workshop on Network Coding, Theory and Applications, pp. 1-6, Hong Kong, China, January 2008.

[7] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," IEEE Transactions on Information Theory, Vol. 49, pp. 371-381, February 2003.

[8] N. Cai and R. W. Yeung, "Network Coding and Error Correction," Proceedings of IEEE Information Theory Workshop, Bangalore, India, pp. 20-25, October 2002.

[9] Z. Zhang, "Linear Network Error Correction Codes in Packet Networks," IEEE Transactions on Information Theory, Vol. 54, pp. 209-218, January 2008.

[10] R. Kotter and F. R. Kschischang, "Coding for Errors and Erasures in Random Network Coding," IEEE Transactions on Information Theory, Vol. 54, pp. 3579-3591, August 2008.

[11] H. Mahdavifar and A. Vardy, "Algebraic List Decoding on the Operator Channel," Proceedings of IEEE International Symposium on Information Theory, Austin, TX, pp. 1193-1197, June 2010.

[12] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using Linear Programming to Decode Binary Linear Codes," IEEE Transactions on Information Theory, Vol. 51, pp. 954-972, March 2005.

# 7.0 MAJOR PUBLICATIONS GENERATED BY IN-HOUSE EFFORT

**The major publications of this research effort are divided into the following research categories: compressive sampling, synchronization, and networking coding.**

## 7.1 Compressive Sampling Publications:

1. L. Shen and B. W. Suter, " Blind 1-bit Compressive Sampling," arXiv preprint, 2013. It is available at http://arxiv.org/abs/1302.1419. Preliminary research was performed in the fall of 2012, with the submission to the journal and arXiv in early 2013.

2. Gao, S. Batalama, D. Pados, and B. W. Suter, "Compressive Sampling with Generalized Polygons," *IEEE Transactions on Signal Processing*, Vol. 58, pp. 4759-4766, 2011.

3. H.-C. Chen, H. T. Kung, D. Vlah, and B. W. Suter, "Measurement Combining and Progressive Reconstruction in Compressive Sensing," *Proceedings of Military Communications Conference*, Baltimore, Maryland, 2011.

## 7.2 Synchronization Publications:

1. I. Sari, E. Serpedin, K.-L. Noh, Q. Chaudhari, and B. W. Suter, "On the Joint Synchronization of Clock Offset and Skew in RBS-Protocol," *IEEE Transactions on Communications*, Vol. 56, pp.700-703, 2008.

2. K.-L. Noh, Y.-C. Wu, K. Qaraqe, and B. W. Suter, "Extension of Pairwise Broadcast Clock Synchronization for Multicluster Sensor Networks," *EURASIP Journal on Advances in Signal Processing*, 2008.

## 7.3 Network Coding Publications:

3. N. Chen, Z. Yan, M. Gadouleau, Y. Wing, and B. W. Suter, "Rank Metric Decoder Architectures for Random Linear Network Coding with Error Control," *IEEE Transactions on VLSI Systems*, Vol. 20, pp.296-309, 2012.

4. Z. Yan, H. Xie, and B. W. Suter, "Rank Deficient Decoding of Linear Network Coding," Proceedings of the IEEE Interantional Conference on Acoustics, Speech, and Signal Processing, Vancouver, British Columbia, Canada, 2013. This paper was submitted in 2012.

5. Z. Yan and B. W. Suter, "Unequal Error Protection and Noncoherent Random Linear Network Coding," *Proceedings of IEEE Conference on Information Sciences and Systems*, Johns Hopkins University, Baltimore, Maryland, 2011.

6. B. W. Suter and Z. Yan U.S. Patent Pending 13/949,319 *Rank Deficient Decoding of Linear Network Coding*. July 2013. A new network coding paradigm to achieve higher throughputs and shorter delays than the full rank decoder. This process will enable source coding at the networking/transport layer. Although this patent was submitted after the completion of the project, the associated provisional patent was submitted in August 2012.

## 8.0 LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

| | |
|---|---|
| **BE** | **Best Effort** |
| **BSR** | **Bit Success Rate** |
| **CN** | **Communication Network** |
| **EF** | **Error Free** |
| **LNC** | **Linear Network Code** |
| **LP** | **Linear Programming** |
| **PSR** | **Packet Success Rate** |
| **RLNC** | **Random Linear Network Code** |
| **VLSI** | **Very Large Scale Integration** |