Vol.2014-MBL-72 No.10
Vol.2014-CDS-11 No.6
2014/8/28

Distributed Coding Schemes for Continuous Data Collection in Wireless Sensor Networks

XIUCAI YE

This manuscript addresses the continuous data collection in WSNs with a mobile Base Station (mBS). We proposed continuous data collection schemes based on distributed coding. Our objective is to provide efficient methods for continuously collecting data segments with a high success ratio.

1. Introduction

Wireless Sensor Networks (WSNs) are composed of a large number of sensor nodes, which do not rely on any pre-deployed network infrastructure. A sensor node has low CPU power, small bandwidth, limited battery and memory storage [1]. Thus, a sensor node can store only a small amount of data collected from its surroundings. A Base Station (BS) collects the data from sensor nodes and functions as an intermediate gateway between the sensor network and the application end users. Note that a BS can be a fixed or mobile one. Recent advances of embedded hardware and robot have made mobile sensors possible [2, 3].

WSNs have a wide range of applications, including environment monitoring, medical care, smart buildings, industrial and military applications [4]. An important problem that arises in application of WSNs is how to collect data continuously, especially in extreme environments. In some extreme environments such as Greenland or Alaska, it is difficult to travel and dangerous to work for humans [5] [6]. Instrumenting the environments with WSNs can enable long-term data collection, which could minimize the exposure of humans while allowing dense, targeted data collection to commence [5].

Consider that data are continuously sensed and collected by the sensor nodes in the extreme environments. The communication between the sensor nodes and a mobile Base Station (mBS) is scarce. Data collection is only performed from time to time by a mBS, as shown in Figure 1. Sensor nodes have to store the continuously collected data segments over time by themselves, and provide the desired data when the mBS arrives and performs data collection. Such kind of data collection is known as continuous data collection [7]. One of the typical examples is the habitat monitoring system in Great Duck Island [8], in which data collection is performed from time to time since seabird colonies are sensitive to human interaction. An efficient data retrieval is usually desired during the data collection.

Due to limited energy and hostile environment, sensor nodes may fail suddenly and unpredictably, resulting in the loss of sensed data. Therefore, to provide robust data retrieval, it is desirable to distribute the sensed data throughout the network for redundant storage [9]. Thus, the mobile base station can retrieve the sensed data from any subset of sensor nodes, even after some sensor nodes have failed.

Coding is a powerful method for data storage and distribution, which can achieve efficient management of redundant data storage [10]. Many data storage and distribution schemes using coding techniques in a centralized way are proposed. A typical coding scheme is the erasure coding [10], in which coding is performed at a central entity and the combined segments are distributed to different storage locations. The Reed-Solomon coding is a well-known erasure coding scheme, which is widely employed in a computer network with distributed storage systems and redundant disk arrays [13, 14]. However, the centralized coding method cannot be employed directly in a sensor network, in which a sensor node is not able to store all the data segments and perform complicated encoding operations alone.

A promising solution is the decentralized coding, which distributes the encoding operations to multiple nodes. Such decentralized coding schemes include decentralized implementations of erasure coding [9] [13], growth coding [14], network coding [7] [15]. Especially, Dimakis et al. [13] proposed an interesting coding scheme called Decentralized Erasure Coding (DEC), which may be applied for WSNs. In the DEC scheme, each sensor node encodes all the collected data segments. However, the DEC scheme is lack of support removing obsolete (old) data, i.e., it cannot support the continuous data collection in which the number of data segments is not predetermined.

Removing obsolete (old) data is another important issue for storing data in a sensor network, since each sensor node has limit storage space. If sensor nodes get unattended from the mBS for a long time (e.g., the bad weather prohibits the mBS from performing data collection for a long time), the total data may exceed the total storage space of the entire sensor network. In many practical applications, new data has higher value than old ones. Thus, a sensor node should be able to remove the old data in order to accommodate newly collected ones [7]. In the DEC scheme [13], removing the old data includes decoding and re-encoding operations, which are time and resource consuming.

Wang et al. [7] proposed an interesting decentralized coding scheme called Partial Network Coding (PNC) for continuous

University of Tsukuba

data collection in a WSN with a mBS. PNC supports removing the obsolete data. Each combined segment encodes only the part of latest original data segments by removing the older data segments. The number of data segments encoded in a combined segment varies from 1 to *m*. By randomly querying a small subset of sensor nodes, the mBS can collect the m latest original data segments from the sensor network, where m is the number of latest original data segments in a time interval t in which $n(t)$ ($m \le n(t)$) data segments are generated. However, not all the m latest original data segments are encoded in each combined segment. That is, the m latest original data segments cannot be always decoded completely when the mBS randomly queries some sensor nodes. Thus, PNC does not have high success ratio of collecting the m latest data segments. The success ratio of data collection in PNC can be improved by extending the storage space in each sensor node and extending the number of sensor nodes queried by the mBS. Note that the storage space of each sensor node depends on the number of latest original data segments. If the number of latest original data segments is large, the overhead for enhancement may be too big for a sensor node.

The organization of the manuscript is shown as follows. Section 2 presents the system model and problem formulation. In Section 3, we present Distributed Separate Coding for Latest Data segment Collection (DSC-LDC) to collect the m latest data segments, where m is the number of latest original data segments in a time interval t in which $n(t)$ ($m \le n(t)$) data segments are generated. In Section r 4, we present Distributed Separate Coding for All Data segment Collection (DSC-ADC) to continuously collect all the $n(t)$ data segments in a time interval t . Section 5 summarizes the manuscript and discusses the direction of the future work.

2. System Description and Problem Formulation

Consider that there are N sensor nodes in a wireless sensor network, where a set of sensor nodes sense information. Each sensor node has *B* buffers, b1, b2, ..., b*B* (i.e., the buffer size of each sensor node is *B*). Each buffer can store only one data segment. Consider to collect the data of samples (e.g., the temperatures measured in the beginning of some fixed time slots) by using a WSN, where the samples are generated continuously. A sample is represented by one data segment c_j, and generated in a fixed time slot. c_j is generated over the j^{th} time slot. c_q is newer than c_p if q > p.

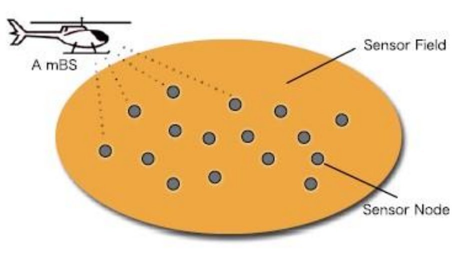

Figure 1. Data collection by a mBS.

Without loss of generality, we consider that there is one mBS (mobile Base Station) which performs data collection from time to time. For example, a helicopter acts as the mBS, as illustrated in Fig 1. During the data collection, the mBS will query a small subset of sensor nodes uniformly at random from the sensor network to collect data.

Consider that the total number of data segments is $n(t)$, where t is the data sensing time interval. Note that t is a variable, which value depends on when the mBS per-forms data collection. $n(t)$ is a non-decreasing function of *t*.

In Section 3, we consider to collect the *m* ($m \le n(t)$) latest original data segments, where m is the number of latest data segments to be collected. Note that, in a time interval t, no matter how many data segments are generated, the required data segments are the m latest original data segments which are generated in the m latest time slots.

In Section 4, we consider to collect all the data segments generated in a data sensing time interval t. The total number of data segments to be collected is $n(t)$. Different form the m late data segment collection, in all data segment collection, the number of collected data segments $n(t)$ is larger and $n(t)$ is a variable that may increase as the time interval t increases.

For data coding, we define a linear function as follows.

$$f_i^u = \sum_{j=1}^{k} \beta_{ij} c_j ,$$

where f_i^u is referred to as a combined segment, which encodes k data segments c_1, ..., c_k ($1 \le k \le n(t)$) in buffer bu of sensor node i. $\overline{\beta_i^u} = (\beta_{i1}, ..., \beta_{ik})$ is a coefficient vector of f_i^u. Each item β_{ij} is randomly generated from a finite field Fq, where q is the finite field size. Note that the size (in bits) of a combined segment f_i^u equals to the size (in bits) of an original data segment c_j. Sensor node i stores the combined segment f_i^u and the associated coefficient vector in buffer bu, instead of storing the *k* original data segments c_1, ..., c_k. Storing the coefficient vector $\overline{\beta_i^u}$ will take an additional storage space of *k* log2(q) bits, which is very small compared to large size of combined segment f_i^u. Take an example similar to that considered in [16], the size of a combined segment is 20 KB, and the size of finite field for coefficients is $q = 2^8$. If f_i^u encodes 10 original data segments, the storage overhead for the coefficient vector is 80 bits or 10 bytes. Thus, the additional storage space required for the coefficient vector is less than 0.05 %, which is a negligible overhead compared to the combined segment.

Since sensor nodes sense the similar environment and collect the data, we assume that the data segments for encoding in each sensor node are the same in a time slot. To achieve this condition, the sensor nodes may also communicate with each other to disseminate data segments. Many methods have been proposed for data dissemination for wireless sensor networks (e.g., the data dissemination methods in [17]). This manuscript focuses on the data encoding and storage processes in each sensor node and the data decoding process in the mBS. We assume that each data segment is recorded by all the sensor nodes by using some existing data dissemination method.

To successfully decode the original data segments, the mBS should access enough number of sensor nodes for the data collection. If the mBS cannot decode the original data segments form the collected combined segments, it will repeat data collection until it succeeds. When the mBS repeats data collection, it randomly accesses enough number of sensor nodes again. The sensor network will consume more energy if the mBS repeats data collection, since more sensor nodes need to upload data to the mBS. Thus, the success ratio of data collection is a major evaluation criterion in the study [7]. We define the success ratio of data collection as follows.

Definition 1 (Success ratio of data collection). The success ratio of data collection is the probability that the mBS successfully collects all the desired original data segments.

3. Latest Data Segment Collection in Wireless Sensor Networks with a Mobile Base Station

In this Section, we study the problem of continuous data collection to collect the m latest data segments, where m is the number of latest data segments in a time interval t in which $n(t)$ ($m \leq n(t)$) data segments are generated.

We present a novel Distributed Separate Coding scheme for m Latest Data segment Collection (DSC-LDC) in wireless sensor networks with a mobile Base Station (mBS). DSC-LDC includes three processes: the data encoding process, the data replacement process and the data decoding process. The data encoding and replacement processes are performed in each sensor node, while the data decoding process is performed in the mBS. We consider two cases. 1) Each sensor node has two buffers (i.e., buffer size $B = 2$). 2) Each sensor node has more than 2 buffers (i.e., buffer size $B > 2$).

3.1 DSC-LDC for the case that each sensor node has two buffers

In the case that each sensor node has two buffers (i.e., buffer size $B = 2$), m -1 original data segments are separately encoded in a combined segment. Let $f_i(r)$ be a combined segment which encodes the r^{th} recorded m-1 original data segments in sensor node i. Generally, we have

$$f_i(r) = \sum_{j=(r-1)(m-1)+1}^{r(m-1)} \beta_{ij} c_j .$$

When a new combined segment $f_i(r)$ is formed, $f_i(r)$ and its associated coefficient vector are stored in a corresponding buffer of sensor node i. A new combined segment encodes the latest original data segments. Note that the mBS wants to collect the m latest original data segments. If there has been a combined segment stored in the corresponding buffer, the new combined segment including the associated coefficient vector will replace the old ones. By the data replacement, each sensor node stores the two latest combined segments, which encode at least m latest original data segments.

We prove that the minimum buffer size for a sensor node in DSC-LDC is two.

When the mBS performs data collection, the set of original data segments encoded in the two latest combined segments in each sensor node includes all the m latest original data segments. In the decoding process, by querying any m-1 sensor nodes, the mBS collects 2(m-1) latest combined segments and the corresponding coefficient vectors. The key property required for successful decoding is that the coefficient vectors are linearly independent. Therefore, the success ratio of data collection in DSC-LDC mainly depends on the probability of linear independence for the coefficient vectors. The probability of linear independency for the coefficient vectors is over 99.6% for $q = 2^8$, and it increases as q increases [7]. Thus, in the case B=2, the success ratio of data collection is very close to 100% by using a large enough finite field size q for coefficients.

3.2 DSC-LDC for the case that each sensor node has more than two buffers

In the case that each sensor node has more than 2 buffers (i.e., buffer size $B > 2$), each sensor node separately encodes a certain number of original data segments in a combined segment. Consider that the certain number of original data segments encoded in a combined segment is x ($x < m$). B buffers can store B combined segments. A new combined segment encodes the latest original data segments. If there has been a combined segment stored in the corresponding buffer, the new combined segment including the associated coefficient vector will replace the old ones. The minimum and optimal value of x is

$$x = \begin{cases} \dfrac{m-1}{B-1} & , m >,. \\ 1 & , m = 1. \end{cases}$$

When the mBS performs data collection, the set of original data segments encoded in the B latest combined segments in each sensor node includes all the m latest original data segments.

In the decoding process, by querying any x sensor nodes, the mBS collects *Bx* latest combined segments and the corresponding coefficient vectors. By solving the B sets of linear equations, the mBS can decode all the original data segments. The key property required for successful decoding is that the coefficient vectors are linearly independent. In the case that $B > 2$, the success ratio of data collection is very close to 100% by using a large enough finite field size q for coefficients.

3.3 Performance evaluation

In DSC-LDC, the necessary storage space in each sensor node can be adjusted by changing the number of sensor nodes queried by the mBS. Furthermore, the transmission cost for data submission to the mBS can be reduced with a few additional storage space in each sensor node.

Note that PNC [7] also addresses the continuous data collection in WSNs. We compare the proposed DSC-LDC scheme with PNC, since it is the existing scheme that has an efficient solution for continuous data collection in WSNs. To the best of our knowledge, PNC is the only one scheme which can support data replacement for continuously collecting data segments.

We compare DSC-LDC with PNC on success ratio of data collection and energy consumption for data transmission. The comprehensive performance evaluation has been conducted through computer simulation. It is shown that the proposed DSC-LDC scheme is the most recommendable one.

3.4 Summary

In this Section, we present Distributed Separate Coding for Latest Data segment Collection (DSC-LDC) in wireless sensor networks with a mobile Base Station (mBS). By separately encoding a certain number of data segments in a combined segment, and doing decoding-free data replacement in the buffers of each sensor node, the proposed DSC-LDC scheme is not only shown as an efficient storage method for continuously collecting data segments, but also achieves a high success ratio of data collection. We show that in DSC-LDC, the number of sensor nodes that should be queried by the mBS can be reduced with a few additional storage space in each sensor node, which result in reducing the energy consumption for data transmission to the mBS. We also show that the success ratio of data collection in DSC-LDC is very close to 100% by using a larger enough finite field for coefficients in both theoretical analysis and simulations.

4. Collecting All Data Continuously in Wireless Sensor Networks with a Mobile Base Station

In the continuous data collection, the sensor nodes may need to collect all the data segments and provide them to the mobile Base Station (mBS). After data collection by the mBS, the end users can try various physical models and test various hypotheses over all the collected data segments.

In this Section, we study the problem of continuous data collection to collect all the $n(t)$ data segments generated in a time interval t. We present Distributed Separate Coding for All Data segment Collection (DSC-ADC) in wireless sensor networks with a mobile base station (mBS).

In DSC-ADC, each sensor node separately encodes a certain number of original data segments in a combined segment and stores it in the corresponding buffer. Let $F_i = \{ f_i^1, f_i^2, ..., f_i^B \}$ be a set of combined segments stored in sensor node i, where f_i^k is stored in buffer *bk*. The associated coefficient vector for f_i^k is also stored in buffer bk. $C(f_i^k)$ is the number of data segments encoded in f_i^k.

We consider two cases. 1) Right arrival case that the mBS arrives on time. 2) Late arrival case that the mBS arrives late. In general condition, the mBS performs data collection in a regular time interval t_0, where $t_0 = \min\{t\}$. This case is called right arrival case. The total number of original data segments generated in time interval t_0 is $n(t_0)$. If the mBS arrives when the time interval $t \geq t_0$, we call this case late arrival case. Note that $n(t_0) \leq n(t)$.

The encoding and decoding processes in the two cases are with some differences. The encoding and decoding processes in the right arrival case are easier, since the total number of original data segments generated in a time interval is a fixed value. While in late arrival case, the total number of original data segments may increase as the time interval increases. A challenge issue is how to let the fixed buffers in each sensor node to store all the data segments whether the mBS arrives on time or late.

4.1 Data encoding and decoding in the right arrival case

In the right arrival case, each sensor node will separately encode a certain number of original data segments in each combined segment, and store the *B* combined segments and the associated coefficient vectors in its *B* buffers. Assume that the certain number of data segments encoded in a combined segment is x ($x < n(t_0)$). Since the maximum number of data segments encoded in a combined segment is x, the total number of data segments encoded in the *B* combined segments is at most Bx. Thus, we can obtain the minimum buffer size of each sensor node as

$$B = \lceil n(t_0) / x \rceil .$$

In the right arrival case, each sensor node with buffer size $B = \lceil n(t_0) / x \rceil$ is enough to store the combined segments which encode all the original data segments.

When the mBS performs data collection, all original data segments are encoded in the B combined segments in each sensor node. In the decoding process, by querying any x sensor nodes, the mBS collects Bx combined segments and the corresponding coefficient vectors. By solving the B sets of linear equations about the B collected combined segments, the mBS can decode all the original data segments. The key property required for successful decoding is that the coefficient vectors are linearly independent. In the right arrival case, the success ratio of data collection is very close to 100% by using a large enough finite field size q for coefficients.

4.2 Data encoding and decoding in the late arrival case

In the late arrival case, if the mBS arrives late, the sensor nodes do not know in which time and how long the mBS will delay, they just separately encode x data segments in a combined segment and store it in the corresponding buffer. If the total number of original data segments $n(t) \le Bx$, by continuing to encode the original data segments in the last combined segment f_i^B, the buffers are still enough. But if $n(t) > Bx$, the data segments will exceed the total storage space of the sensor nodes if they still combine x original data segments in a combined fashion.

The encoding and storage process when $n(t) \le Bx$ are the same as right arrival case. When $n(t) > Bx$, the original data segments will be encoded one by one in the existing combined segments. Let r be a positive integer. If the sequence number of c_j satisfies $j = r \cdot Bx + k$, c_j will be encoded in f_i^k as the following equation.

$$f_i^k = f_i^{k'} + \beta_{ij} c_j ,$$

where $f_i^{k'}$ is the former combined segment stored in buffer k before c_j is encoded.

When $n(t) \le Bx$, the mBS can reconstruct all the original data segments by querying any x sensor nodes with high probability. The decoding process is the same as the right arrival case. When $n(t) > Bx$, the mBS queries any $x + v$ sensor nodes to collect data, where $v = \lceil (n(t) \text{ -} Bx) / B \rceil$. When the mBS performs data collection, there are B combined segments stored in each sensor node. After decoding all the linear equations about the B combined data segments, the mBS can obtain all the original data segments. The success ratio of data collection is very close to 100% by using a large enough finite field size q for coefficients.

4.3 Performance evaluation

We evaluate the performance of the proposed DSC-ADC scheme by simulations. In the simulations, we consider the following scenarios. 1) Success ratio of data collection. 2) Energy consumption during data collection by the mBS. 3) Data collection time.

We evaluate the proposed scheme by changing the buffer size in each sensor node. The number of sensor nodes that should be queried by the mBS can be reduced with a few additional storage space in each sensor node. Thus, the energy consumption and data collection time can be reduced with a few additional storage space in each sensor node. The simulation results further demonstrates the feasibility and superiority of the proposed DSC-ADC scheme.

4.4 Summary

In this Section, we present Distributed Separate Coding for All Data segment Collection (DSC-ADC) in wireless sensor networks with a mobile base station (mBS). By separately encoding a certain number of data segments in a combined segment, and storing the combined segments in the corresponding buffers of each sensor node, the DSC-ADC scheme provides an efficient storage method to collect all data segments that can be applied in both the right arrival case and the late arrival case. By randomly querying a small subset of sensor nodes, the mBS can reconstruct all the original data segments with high probability. We also show that the success ratio of data collection in DSC-ADC is very close to 100% by using a larger enough finite field for coefficients in both theoretical analysis and simulations.

5. Conclusion and Future Work

In this manuscript, we address the continuous data collection in wireless sensor networks with a mobile base station. We consider two scenarios of continuous data collection. 1) Latest data segment collection. 2) All data segment collection. We propose two Distributed Separate Coding based schemes for the two scenarios, respectively. The two proposed schemes are Distributed Separate Coding for Latest Data segment Collection (DSC-LDC) and Distributed Separate Coding for All Data segment Collection (DSC-ADC).

In DSC-LDC, by separate encoding and doing decoding-free data replacement in the buffers of each sensor node, the proposed DSC-LDC scheme is an efficient method to collect the m latest data segments with high success ratio. DSC-LDC is flexible and efficient compared to the related work (i.e., PNC). The discussion and simulation both show that DSC-LDC improves the performance in many situations. In DSC-ADC, by separate encoding and strategic storage in the buffers of each sensor node, the DSC-ADC scheme provides an efficient storage method to collect all data segments with high success ratio in both the right arrival case and the late arrival case. We show that the success ratio of data collection in DSC-LDC and DSC-ADC are both very close to 100% by using larger enough finite field for coefficients in both theoretical analysis and simulations.

In this manuscript, the proposed distributed coding schemes are based on random linear coding, since random linear coding is easy and suitable to deploy in wireless sensor networks. In the future work, we will consider other coding methods, such as fountain coding. Fountain coding is a promising solution to reduce the decoding complexity. However, the implement of Fountain codes in WSNs is more difficult than that of random linear coding. We will consider an efficient scheme to strategically encode and store the sensed data by Fountain Coding in wireless sensor networks.

Reference

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. Wireless Sensor Networks: A Survey. Comm. ACM 38, 4 (2002), 393–422.

[2] M. Di Francesco, S. K. Das, and G. Anastasi. 2011. Data Collection inWireless Sensor Networks with Mobile Elements: A Survey. ACM Transactions on Sensor Networks 8, 7 (2011), 7–31.

[3] D. Wang, J. Liu, and Q. Zhang. 2013. On Mobile Sensor Assisted Field Coverage. ACM Transactions on Sensor Networks 9, 3 (2013).

[4] D. Culler, D. Estrin, and M. Srivastava. 2004. Overview of Sensor Networks. IEEE Comput. 37, 8 (Special Issue on Sensor Networks) (2004), 41–49.

[5] S. Williams, M. Hurst, and A. M. Howard. 2010. Development of a Mobile Arctic Sensor Node for Earth-Science Data Collection Applications. In in American Institute of Aeronautics and Astronautics, Infotech@ Aerospace,(Atlanta, GA).

[6] G. Tolle 2005. Sonoma redwoods data (2005). http://www.cs.berkeley.edu/ get/sonoma.

[7] D. Wang, Q. Zhang, and J. Liu. 2008. Partial Network Coding: Concept, Performance, and Application for Continuous Data Collection in Sensor Networks. ACM Transactions on Sensor Networks 4, 3 (2008), 1–22.

[8] S. Williams, M. Hurst, and A. M. Howard. 2010. Development of a Mobile Arctic Sensor Node for Earth- Science Data Collection Applications. In in American Institute of Aeronautics and Astronautics, Infotech@ Aerospace,(Atlanta, GA).

[9] A. Dimakis, V. Prabhakarna, and K. Ramchandran. 2005. Ubiquitous access to distributed data in largescale sensor networks through decentralized erasure codes. In Proceeding of of the International Conference on Information Processing in Sensor Networks (IPSN05). ACM, 111–117.

[10] S. Lin and D. Costello. 2004. Error Control Coding: Fundamentals and Applications. Prentice Hall, Upper Saddle River, NJ.

[11] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. 2000. Oceanstore: An architecture for global-scale persistent storage. In Proceedings of IACM ASPLOS 2012. ACM.

[12] W. A. Burkhard and J. Menon. 1993. Disk array storage system reliability. In Proceedings of the 23rd Intenrational Symposium on Fault-Tolerant Computing. IEEE.

[13] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran. 2006. Decentralized erasure codes for distributed networked storage. IEEE Trans. Info. Theory 52, 6 (2006), 2809–2816.

[14] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein. 2006. Growth codes: Maximizing sensor network data persistence. In Proceedings of ACM Sigcom 06.

[15] A. G. Dimakis, P. B. Godfrey, Yunnan Wu, M. J. Wainwright, and K. Ramchandran. 2010. Network coding for distributed storage systems. IEEE Trans. Info. Theory 56, 9 (2010), 4539–4551.

[16] S. Acedanski, S. Deb, M. Medard, and R. Koetter. 2005. How good is random linear coding based distributed networked storage. In Proceedings of First Workshop Network Coding, Theory, and Applications (Net- Cod05). Riva del Garda, Italy.

[17] Z. Kong, S. Aly, and E. Soljanin. 2010. Decentralized coding algorithms for distributed storage in wireless sensor networks. IEEE Journal on Selected Areas in Communications 28 (2010), 261 – 267.