



A Dependable Slepian-Wolf Coding Based Clustering Algorithm for Data Aggregation in Wireless Sensor Networks

Qihang Shu, Qiang Hu, Jun Zheng, Nathalie Mitton

► To cite this version:

Qihang Shu, Qiang Hu, Jun Zheng, Nathalie Mitton. A Dependable Slepian-Wolf Coding Based Clustering Algorithm for Data Aggregation in Wireless Sensor Networks. International Conference on Wireless Communications and Signal Processing (WCSP), Oct 2013, Hangzhou, China. hal-00869426

HAL Id: hal-00869426

<https://inria.hal.science/hal-00869426>

Submitted on 2 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Dependable Slepian-Wolf Coding Based Clustering Algorithm for Data Aggregation in Wireless Sensor Networks

Qihang Shu¹, Qiang Hu¹, Jun Zheng¹, and Nathalie Mitton²

¹National Mobile Communications Research Laboratory
Southeast University

Nanjing, Jiangsu 210096, China

²Inria Lille – Nord Europe
40 Avenue Halley, 59650 Villeneuve D'ASCQ, France
Email: junzheng@seu.edu.cn

Abstract—This paper considers the Slepian-Wolf coding based data aggregation problem and the corresponding dependable clustering problem in wireless sensor networks (WSNs). A dependable Slepian-Wolf coding based clustering (D-SWC) algorithm is proposed to provide dependable clustering against cluster-head failures. The proposed D-SWC algorithm attempts to elect a primary cluster head and a backup cluster head for each cluster member during clustering so that once a failure occurs to the primary cluster head the cluster members within the failed cluster can promptly switch over to the backup cluster head and thus recover the connectivity of the failed cluster to the data sink without waiting for the next-round clustering to be performed. Simulation results show that the D-SWC algorithm can effectively increase the amount of data transmitted to the data sink as compared with an existing non-dependable clustering algorithm for Slepian-Wolf coding based data aggregation in WSNs.

Key words—Slepian-Wolf coding; dependable clustering; data aggregation; wireless sensor network

I. INTRODUCTION

In a wireless sensor network (WSN), a number of sensor nodes are densely deployed in an area of interest with a data sink deployed within or near the area^[1]. The data observed by the sensor nodes usually have a certain spatial correlation and the degree of the spatial correlation increases as the distance between sensor nodes decreases^[2]. The spatial correlation can lead to considerable data redundancy in the network and transmitting redundant data would cause unnecessary energy consumption. To increase energy efficiency and prolong network lifetime, it is desirable to perform in-network data aggregation to remove the data redundancy in the network.

Slepian-Wolf coding^[3-4] is a distributed source coding (DSC), which can remove data redundancy without communication between sensor nodes. To take advantages of

the properties of Slepian-Wolf coding, sensor nodes must know the correlation structure of the entire network *a priori* and the correlation structure of the network depends on the characteristics of the observed phenomenon and the distances between the sensor nodes in the network^[4]. However, applying Slepian-Wolf coding to a large network in a global manner is usually difficult because in that case every sensor node needs to know the correlation structure of the whole network, which would incur much additional control overhead.

An effective way to address the above issue is to introduce node clustering in a WSN to improve energy efficiency and prolong network lifetime^[5]. This technique can help reduce the energy consumption of sensor nodes by allowing cluster members in a cluster to send data to their associated cluster head at a short distance and the cluster head to transmit locally aggregated data to the data sink at a long distance. Due to this fact, node clustering can be effectively combined with Slepian-Wolf coding to increase the energy efficiency in data aggregation. Using clustering, a network is configured into a certain number of clusters. Each cluster contains a smaller number of neighbor sensor nodes. In this case, each cluster node only needs to know the local correlation structure of the cluster when applying Slepian-Wolf coding to a large network, which is more feasible.

One concern that arises in node clustering is the cluster-head failure problem caused by physical damage or external attack, which would result in the loss of connectivity of all affected cluster members in a failed cluster, and disrupt the data communication within the cluster and even the normal operation of the entire network. To address this problem, it is desirable to find an efficient way to recover the connectivity of the affected cluster nodes in the event of a cluster-head failure. A traditional way to this end is to re-cluster the sensor nodes once a failure occurs^[6-7], but such re-clustering is usually time-insensitive, and would disrupt the normal data communication of the affected sensor nodes during re-clustering and lead to incomplete data transmission in the network.

In this paper, we consider Slepian-Wolf coding based data aggregation problem and the corresponding dependable

This work was supported by the National Natural Science Foundation of China under Grant No. 61071115, the Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20110092110007, and the Research Fund of National Mobile Communications Research Laboratory, Southeast University, China, under Grant No. 2012A02.

clustering problem in WSNs. A dependable Slepian-Wolf coding based clustering (D-SWC) algorithm is proposed to provide survivable clustering against cluster-head failures, which elects a couple of cluster heads for each cluster member: a primary head and a backup head. Once a failure occurs to the primary cluster head, the cluster members within the failed cluster can promptly switch over to the backup cluster head and thus recover the connectivity of the failed cluster to the data sink without waiting for the next-round clustering to be performed. Simulation results show that the D-SWC algorithm can effectively increase the amount of data transmitted to the data sink as compared with an existing non-dependable clustering algorithm for Slepian-Wolf coding based data aggregation in WSNs.

The rest of the paper is organized as follows. Section II describes the dependable clustering problem considered in this paper. Section III presents the proposed D-SWC algorithm. In Section IV evaluates the performance of the D-SWC algorithm through simulation results. Section V concludes this paper.

II. PROBLEM STATEMENT

This section first introduces the concept of Slepian-Wolf coding, and then describes the dependable clustering problem and review related work.

A. Slepian-Wolf Coding

Assume that a network has N sensor nodes uniformly distributed in an area of interest and each node i produces reading X_i . All the readings form a set of jointly ergodic sources, denoted by $\mathbf{X} = (X_1, X_2, \dots, X_N)$. These readings follow a distribution of $p(x_1, x_2, \dots, x_N)$, which corresponds to the spatial correlation structure known by each node *a priori*. According to the Slepian-Wolf Theorem^[3], the sensor nodes can jointly encode their data without inter-node communication with an overall rate (in bits) lower-bounded by their joint entropy $H(X_1, X_2, \dots, X_N)$ provided that their respective rates meet the following condition:

$$R(\mathbf{G}) \geq H[\mathbf{X}(\mathbf{G}) | \mathbf{X}(\mathbf{G}^c)]$$

for all $\mathbf{G} \in \{1, 2, \dots, N\}$, where $\{1, 2, \dots, N\}$ is a set of the indices of sensor nodes in the network, \mathbf{G} is a set of sensor nodes, \mathbf{G}^c is the complementary set of \mathbf{G} , $H(\mathbf{X})$ is the entropy of \mathbf{X} , and

$$R(\mathbf{G}) = \sum_{i \in \mathbf{G}} R_i, \quad X(\mathbf{G}) = \{X_j, j \in \mathbf{G}\}.$$

According to chain theory^[2], for any order of N nodes, there always exists a rate allocation $\{R_i\}_{i=1}^N$ which makes the total number of bits generated by all nodes equal to the value of their joint entropy, i.e.,

$$\sum_{i=1}^N R_i = H(X_1, X_2, \dots, X_N),$$

where

$$R_1 = H(X_1)$$

$$R_i = H(X_i / X_{i-1}, X_{i-2}, \dots, X_1), \quad 2 \leq i \leq N$$

Therefore, all sensor nodes in a cluster, say cluster \mathbf{A} , can encode their data with $H(X_1, X_2, \dots, X_{|\mathbf{A}|})$ bits using Slepian-Wolf coding without inter-node communication, and there always exists an optimal rate allocation which can achieve this local maximum compression gain.

B. Clustered Slepian-Wolf Coding

Assume that a network consists of a set of sensor nodes \mathbf{V} . Each sensor node is initially a cluster-head candidate. The diameter of each candidate is equal, and all other nodes within the diameter can become its cluster members. The nodes within the cluster diameter of candidate v constitutes a finite point set Ω_v with the cardinality of $|\Omega_v|$, which is called the neighbor set of candidate v . $\mathbf{P}(\Omega_v)$ denotes the power set of Ω_v , which is a set whose elements are the subsets of Ω_v , and it constitutes all possible combinations of nodes in Ω_v .

Therefore, the cardinality of $\mathbf{P}(\Omega_v)$ is $2^{|\Omega_v|}$. Since a candidate v associated with each combination of nodes (cluster members) within its cluster diameter (e.g., a set of nodes \mathbf{A}_v , where $\mathbf{A}_v \in \mathbf{P}(\Omega_v)$), can form a unique potential cluster (e.g., $\mathbf{A} := \mathbf{A}_v \cup \{v\}$), it can generate up to $2^{|\Omega_v|}$ potential clusters^[7]. Further, since every node is initially a candidate, the network consists of $|\mathbf{V}|$ candidates. Therefore, the whole network consists of a cluster set \mathbf{S} , which includes $\sum_{v \in \mathbf{V}} 2^{|\Omega_v|}$ potential clusters. At the same time, we can encode each potential cluster \mathbf{A} with $H(X_1, X_2, \dots, X_{|\mathbf{A}|})$ using Slepian-Wolf coding in that cluster.

Under the above assumptions, the clustered Slepian-Wolf coding problem is to select a set of disjoint potential clusters \mathbf{C}^* from the cluster set \mathbf{S} to cover the whole network such that the global compression gain of Slepian-Wolf coding is maximized, or the total rate (bits) of the encoded data generated by all the clusters in the network is minimized^[7], i.e.,

$$\mathbf{C}^* = \arg \min_{\mathbf{C} \subseteq \mathbf{S}} \sum_{\mathbf{A} \in \mathbf{C}} H(X(\mathbf{A})),$$

where

$$\bigcup_{\mathbf{A} \in \mathbf{C}^*} \mathbf{A} = \mathbf{V},$$

$$\bigcap_{\mathbf{A} \in \mathbf{C}^*} \mathbf{A} = \emptyset,$$

$$X(\mathbf{A}) = \{X_j, j \in \mathbf{A}\},$$

and \mathbf{C} denotes a set of disjoint clusters from the cluster set \mathbf{S} to cover the whole network.

C. Dependable Clustering

We consider the dependable clustering problem, i.e., construct a cluster hierarchy with each cluster having two

different cluster heads: a primary head and a backup head. This problem is similar to the *domatic* partition problem in graph theory^[8], where a *domatic* partition is a partition of vertices in which each part is a dominating set. With the *domatic* partition, each vertex in the graph is either in the dominating set or has a neighbor in the set. In a clustered network, each sensor node is either a cluster head or a cluster member. Therefore, all cluster heads in the network actually constructs a *dominating set* and the *domatic* partition can generate several different *dominating sets*. Correspondingly, a solution to the *domatic* partition problem can also partition the network such that each cluster member can be covered by several different sets of cluster heads. The *domatic* partition problem is a well-known NP-complete problem. A simple way to solve the problem is to greedily select small dominating sets and iteratively remove the selected dominating sets from the graph until the remainder is no longer dominating. However, the greedy algorithm aims to find multiple backup cluster heads, while in this work we only consider one primary cluster head and one backup cluster head for solving the dependable clustering problem.

D. Related Work

Slepian-Wolf coding based data aggregation has been studied in the literature^[9-12]. In [9-10], global Slepian-Wolf coding and shortest-path routing are studied with an aim to minimize the total cost for sending compressed data. The results show that applying Slepian-Wolf coding globally is difficult. However, this work does not consider distributed clustering with Slepian-Wolf coding. In [11], Marco and Neuhoff studied the application of Slepian-Wolf coding locally within each cluster and showed its advantage in overcoming the effect of node failures on the data reconstruction at the remote sink. However, this work does not propose any clustering protocol to construct a cluster hierarchy. In [12], the clustered Slepian-Wolf coding problem is studied and a distributed optimal compression clustering protocol (DOC²) is proposed to solve the problem. On the other hand, most existing clustering protocols for WSNs do not consider correlation structure and thus cannot take advantage of Slepian-Wolf coding to increase the efficiency of data aggregation.

In the context of dependable clustering, there is not much related work reported in the literature. In [6-7], Mhatre and Heinzelman et al. introduced periodic re-clustering to address the problem. However, this would disrupt the normal communication in the failed cluster. In [13], Younis et al. proposed a robust energy-efficient distributed clustering (REED) for WSNs, which can construct a robust clustered hierarchy by selecting k independent sets of cluster heads. In [14], Gupta and Younis proposed a fault-tolerant clustering algorithm to perform re-clustering locally once most cluster heads reach a consensus about the occurrence of a cluster-head failure. In [15], Wang et al. proposed a dependable clustering protocol to solve the cluster-head failure problem in underwater sensor networks (UWSNs). However, these clustering algorithms do not consider Slepian-Wolf coding for data aggregation.

III. DEPENDABLE SLEPIAN-WOLF CODEING BASED CLUSTERING ALGORITHM

In this section, we present the proposed dependable Slepian-Wolf coding based clustering (D-SWC) algorithm for data aggregation in WSNs.

A. Overview

The D-SWC algorithm is based on the greedy algorithm proposed in [16] for solving the *domatic* partition problem and the distributed optimal-compression clustering protocol (DOC²) for maximizing the amount of transmitted data in [12]. The objective of the D-SWC algorithm is to elect a primary cluster head and a backup cluster head for each cluster member during clustering, taking into account Slepian-Wolf coding. Compared with the greedy algorithm, the D-SWC algorithm has two major differences:

- 1) The greedy algorithm must ensure that multiple different sets of cluster heads be found so that each cluster member can be associated with multiple backup cluster heads. This makes it difficult to consider energy efficiency in clustering. For the D-SWC algorithm, however, it only needs to elect one primary cluster head and one backup cluster head, which makes it possible to consider energy efficiency in clustering.
- 2) The greedy algorithm must find a new dominating set or a set of backup cluster heads that cover not only the remaining cluster members but also the primary cluster heads. For the D-SWC algorithm, the elected backup cluster heads only need to cover the remaining cluster members because they become cluster heads only when the primary cluster heads fail.

B. Procedures

The D-SWC algorithm can be divided into three phases: initialization, clustering, and finalization.

a) Initialization phase

In the initialization stage, each node has three possible states: cluster head (or *head*), cluster member (or *memb*) and cluster-head candidate (or *cand*). Initially, each node performs neighbor discovery to discover its one-hop neighbors and maintain an uncovered neighbor set, which contains its one-hop neighbors still in the *cand* state. A candidate can potentially generate a number of different clusters by combining different nodes in its uncovered neighbor set. If a candidate v has an uncovered neighbor set \mathbf{U}_v , the number of potential clusters the candidate can generate is $2^{|\mathbf{U}_v|}$. Once the uncovered neighbor set of a candidate is determined, the candidate will select a cluster with the minimum average joint entropy as its qualified cluster from all the potential clusters that can be generated from the uncovered neighbor set.

For example, consider a cluster-head candidate v with a neighbor set \mathbf{Q}_v . The power set of \mathbf{Q}_v , $\mathbf{P}(\mathbf{Q}_v)$, constitutes all possible combinations of nodes in \mathbf{Q}_v . Assume that \mathbf{M}_v is a cluster member set of candidate v . Our objective is to select a potential cluster with the minimum average entropy as the

representative cluster of v , which is denoted by A_v and satisfies the following conditions, i.e.,

$$\begin{aligned} \mathbf{M}_v &\leftarrow \arg \min_{\mathbf{M}_v \subseteq P(\Omega_v)} \left\{ \frac{H(X(\{v\} \cup \mathbf{M}_v))}{|\{v\} \cup \mathbf{M}_v|} \right\} \\ \mathbf{A}_v &\leftarrow \{v\} \cup \mathbf{M}_v \\ \mathbf{C}^* &= \arg \min_{\mathbf{C} \in \mathcal{S}} \sum_{A_v \in \mathbf{C}} H(X(A_v)) \end{aligned}$$

where $H(\{v\} \cup \mathbf{M}_v)$ represents the joint entropy of the cluster, and $H(\{v\} \cup \mathbf{M}_v)/|\{v\} \cup \mathbf{M}_v|$ is considered as the average entropy of the cluster.

b) Clustering phase

In the clustering phase, each candidate v first performs the following procedures:

- 1) Construct its own potential clusters by comparing every combination of the neighbors in its one-hop uncovered neighbor set;
- 2) Calculate the average entropy of each potential cluster based on the distances between the nodes within the cluster;
- 3) Select the representative cluster with the maximum global compression gain among all the potential clusters;
- 4) Send the average entropy of its qualified cluster to all candidates within its 2-hop range.

After that, each candidate compares its own entropy with the average entropy received from other candidates within its 2-hop range. If the candidate itself has the minimum average entropy, it will become a cluster head. In this case, it will invite all the nodes in its representative cluster to join its cluster by advertising an *INVITE* message. If a candidate receives an *INVITE* message destined to it, it will perform the following procedures:

- 1) Change its status from a cluster-head candidate to a cluster member;
- 2) Extract the cluster-head ID from the *INVITE* message;
- 3) Record the node sending the *INVITE* message as the primary cluster head and add the node's ID into its cluster-head (CH) list;
- 4) Broadcast a *JOIN* message to all the nodes within its cluster diameter to acknowledge the receipt of the *INVITE* message and to notify the other candidates within the cluster diameter that it has become a cluster member of some cluster head.

If a candidate does not receive any *INVITE* message, it will stay in the candidate status and reselect its representative cluster because at this time some neighbors in its uncovered neighbor set may have become cluster heads or cluster members of other cluster heads. Even if a node has become a cluster member, it still needs to receive the *INVITE* messages and add the senders' IDs extracted from the messages into its cluster-head list. The above procedures are performed by all candidates until they all become either a cluster head or a cluster member. At the end of the clustering phase, there will be no candidate in the network and the cluster-head list of

every cluster member will contain not only a primary cluster head, but also one or more backup cluster heads.

c) Finalization phase

In the clustering phase, the cluster-head list generated may contain one or more backup cluster heads except the primary cluster head. In this case, the one with the shortest distance to a cluster member is selected as the backup cluster head for that cluster member in the finalization phase.

The D-SWC Algorithm

```

status(v) ∈ {head, cand, memb} ;
status(v) ← cand ;
send and receive status within 1 hop;
U_v ← {u | status(u) = cand, u ∈ Ω_v} ;
while status(u) ≠ head do {
    if status(u) = memb then
        while v receives INVITE(u, X_u) do
            P_List(v) ← {u} ∪ P_List(v);
    else
        C_v ← P(U_v);
        M_v ← arg min_{M_v ⊆ C_v} {H(X(\{v\} ∪ M_v))} / |\{v\} ∪ M_v| ;
        A_v ← {v} ∪ M_v;
        H_avg(A_v) ← H(X(A_v)) / |A_v| ;
        send H_avg(A_v) within 2 hops;
        Z ← {u | H_avg(A_v) sent by u is received} ;
        if H_avg(A_v) = min_{u ∈ Z} {H_avg(A_v)} then
            {Let status(v) ← head ;
            send INVITE(v, X_v) within 1 hop;}
        else wait until selection timeout
        if v receives INVITE(u, X_u) then
            if v ∈ X_u then
                {Let status(v) ← memb;
                Cluster_head(v) ← u;
                P_List(v) ← {u} ∪ P_List(v);
                send JOIN(v, u);}
            else {Let status(v) = cand ;
            U_v ← U_v - {u};}
        else if v receives JOIN(u, w) then
            {Let status(v) ← cand ;
            U_v ← U_v - {u};}
    }
    wait until clustering timeout
    if status(u) = memb then
        {B_List(v) ← P_List(v) - Cluster_head(v);
        Backup_head(v) = arg min_{u ∈ B_List(v)} {dist(v, u)} ;}

```

Fig. 1 The pseudo codes of the D-SWC algorithm.

Figure 1 gives the pseudo codes of the D-SWC algorithm, the notations used in the pseudo codes are explained as follows:

- Ω_v : a finite point set within one hop of candidate v .
- $P(\Omega_v)$: a set whose elements are the subsets of Ω_v .
- C_v : all the possible combinations of nodes within one-hop range of candidate v .
- M_v : the set containing the representative cluster members of candidate v .
- A_v : the representative cluster of candidate v ;
- X_v : a set containing the cluster members of A_v ;
- $H_{avg}(A_v)$: the average entropy of A_v ;
- Z : a set containing the average entropy sent by other cluster heads within the 2-hop range of a candidate;
- $P_List(v)$: the primary cluster-head list of candidate v ,
- $B_List(v)$: the backup cluster-head list of candidate v ;
- $Cluster_head(v)$: the primary cluster head of candidate v .
- $Backup_head(v)$: the backup cluster head of candidate v .
- $INVITE(v, X_v)$: a message used to invite the nodes in set X_v to become the cluster members of candidate v ;
- $JOIN(v, u)$: a message used to confirm that node v has received the $INVITE$ message sent by candidate u and has joined the cluster of candidate u .

IV. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed D-SWC algorithm through simulation results. For evaluation, we compare the D-SWC algorithm and the DOC² algorithm in [12] and the performance metrics include the amount of data transmitted to the data sink and the average residual energy of sensor nodes in the network.

In the simulation experiments, we used a network with 100 sensor nodes uniformly deployed in a region of 100m×100m. The cluster diameter is 15m. For the correlation structure, we use an approximation algorithm to calculate the amount of the joint entropy generated by a cluster with N nodes^{[4][10]}. We assume that the observations X_1, X_2, \dots, X_N generated by N sensor nodes can be modeled as an N -dimensional jointly Gaussian model with mean $(0, 0, \dots, 0)$ and covariance matrix K . The density of X is

$$f(X) = \frac{1}{(\sqrt{2\pi})^N |K|^{1/2}} e^{-\frac{1}{2} X^T K^{-1} X}$$

and the differential entropy of (X_1, X_2, \dots, X_N) is

$$h(X_1, X_2, \dots, X_N) = \frac{1}{2} \log(2\pi e)^N |K| \text{ bits},$$

where $|K|$ denotes the determinant of matrix K ^[4]. For matrix K , we use an exponential model of the covariance $k_{ij} = \exp(-d_{ij}\theta)^2$ in [10] to model its elements, where d_{ij} represents the distance between the nodes X_i and X_j , and θ indicates the extent of correlation. For the sake of simplicity and without loss of generality, we apply the differential entropy to replace the discrete entropy because the differential entropy differs from the discrete entropy by a constant^{[10][15]}.

Figure 2 shows the total amount of data received by the data sink with D-SWC and DOC², respectively, within 10 simulation rounds. To investigate the impact of a cluster-head failure on the performance, four cluster-head failures are generated in each round, and the failed cluster heads are randomly selected from the primary cluster heads. It is seen that D-SWC transmits more data to the sink than DOC² during the 10 simulation rounds. This is because in the event of a cluster-head failure, D-SWC allows the cluster members within a failed cluster to transmit their data to a backup cluster head, while with DOC² the nodes have to wait until the next-round clustering is completed to transmit their data.

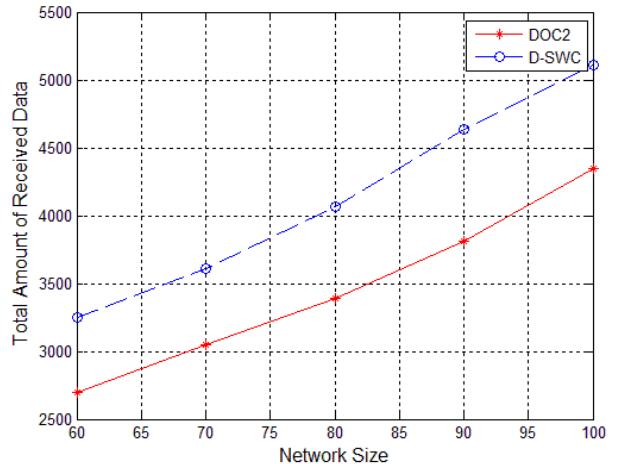


Fig.2 Total amount of data received by the data sink.

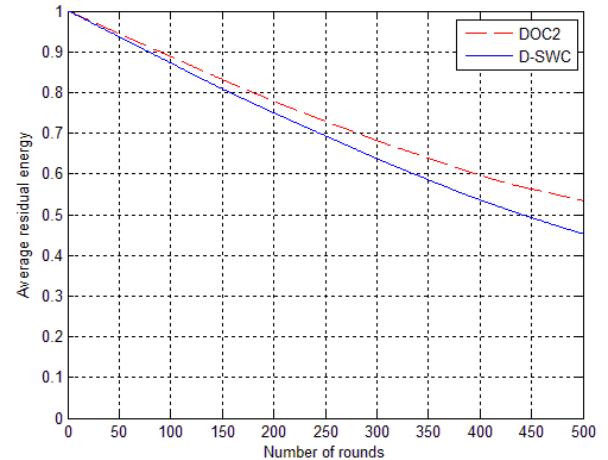


Fig.3 Average residual energy of sensor nodes in the network.

Figure 3 shows the average residual energy of sensor nodes in the network with the D-SWC algorithm and the DOC² algorithm, respectively. It is seen that with the D-SWC algorithm the nodes consume a little more energy than that with the DOC² algorithm during the simulations. This is because with the DOC² algorithm the sensor nodes within a failed cluster do not transmit data until the next-round re-clustering is completed, while with the D-SWC algorithm the nodes can continue to transmit data to the backup cluster head, thus consuming more energy in the same simulation time. It should be pointed out that the consumed more energy is used for data transmission, not a waste.

V. CONCLUSIONS

In this paper, we proposed a dependable Slepian-Wolf coding based clustering (D-SWC) algorithm to solve the dependable clustering problem in Slepian-Wolf coding based data aggregation for WSNs. The D-SWC algorithm attempts to elect a primary cluster head and a backup cluster head for each cluster member during clustering so that once a failure occurs to the primary cluster head, the cluster members within the failed cluster can promptly switch over to the backup cluster head and thus recover the connectivity of the failed cluster to the data sink without waiting for the next-round clustering to be performed. The simulation results have shown that the D-SWC algorithm can effectively increase the amount of data transmitted to the data sink as compared with the DOC² algorithm. In future work, we will study the cluster-head switch over problem in the event of a cluster-head failure for supporting the D-SWC algorithm.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks (Elsevier) Journal*, vol. 4, no. 12, Mar. 2002, pp. 393–422.
- [2] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: Theory and applications for wireless sensor networks," *Computer Networks Journal (Elsevier)*, vol. 45, no. 3, 2006, pp. 245–261.
- [3] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, July 2003, pp. 471–480.
- [4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: John Wiley and Sons, Inc., 1991.
- [5] Jun Zheng and Abbas Jamalipour, *Wireless Sensor Networks: A Networking Perspective*, Wiley-IEEE Press, ISBN: 9780470167632, Sept. 2009.
- [6] V. Mhatre and C. Rosenberg, "Homogeneous vs heterogeneous clustered networks: A comparative study," in Proc. of IEEE ICC'04, vol. 6, Paris, France, Jun. 2004, pp. 3646–3651.
- [7] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *Wireless Communications, IEEE Transactions on*, vol. 1, no. 4, Oct. 2002, pp. 660–670.
- [8] U. Feige, M. Halldórsson, G. Kortsarz, and A. Srinivasan, "Approximating the Domatic Number," *SIAM Journal on Computing*, vol. 32, no. 1, Jan. 2003, pp. 172–195.
- [9] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked slepianwolf: theory, algorithms, and scaling laws," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, 2005, pp. 4057–4073.
- [10]R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in Proc. of INFOCOM'04, vol. 4, Hong Kong, Mar. 2004, pp. 2571–2582.
- [11]D. Marco and D. L. Neuhoff, "Reliability vs. efficiency in distributed source coding for field-gathering sensor networks," in Proc. of the Third International Symposium on Information Processing in Sensor Networks (IPSN'04), Berkeley, CA Apr. 26–27, 2004, pp. 161–168.
- [12]P. Wang, C. Li, and J. Zheng, "Distributed data aggregation using clustered Slepian-Wolf coding in wireless sensor networks," in Proc. of IEEE ICC'07, Glasgow, UK, June 2007, pp. 3616–3622.
- [13]O. Younis, S. Fahmy, and P. Santi, "An Architecture for Robust Sensor Network Communications", *International Journal of Distributed Sensor Networks*, vol. 1, no. 3-4, 2005, pp. 305–327.
- [14]G. Gupta. and M. Younis, "Fault-tolerant clustering of wireless sensor networks," in Proc. of IEEE WCNC'03, vol. 3, Mar. 2003, pp. 1579 – 1584.
- [15]P. Wang, C. Li, J. Zheng, H. Mouftah, "A Dependable Clustering Protocol for Survivable Underwater Sensor Networks," in Proc. of the IEEE ICC'08, Beijing, China, May 2008, pp. 3263–3268.
- [16]Chvatal, "A greedy heuristic for the set covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, 1979, pp. 126–135.