

Research Article

Data Dissemination in Wireless Sensor Networks with Network Coding

Xiumin Wang,^{1,2} Jianping Wang,² and Yinlong Xu¹

¹School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, 230027, China

²Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

Correspondence should be addressed to Jianping Wang, jianwang@cityu.edu.hk

Received 28 September 2009; Revised 5 March 2010; Accepted 25 April 2010

Academic Editor: Christina Fragouli

Copyright © 2010 Xiumin Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In wireless sensor networks (WSNs), it is often necessary to update the software running on sensors, which requires reliable dissemination of large data objects to each sensor with energy efficiency. During data dissemination, due to sleep scheduling designed for energy efficiency, some sensors may not receive some packets at some time slots. In the meantime, due to the unreliability of wireless communication, a sensor may not successfully receive a packet even when it is in the active mode. Thus, retransmission of such packets to those sensors is necessary, which consumes more energy and increases the delay of data dissemination cycle. In this paper, we propose a network coding-based approach in data dissemination such that data dissemination can be accomplished at the earliest time. Thus, less energy is consumed and the delay can be decreased. The impact of packet loss probability and the sleep probability of sensors on the network coding gain is analyzed. A threshold is also given to decide whether the current sleep scheduling is effective on energy saving in data dissemination process or not. Simulation results demonstrate the effectiveness and scalability of the proposed work.

1. Introduction

Recently, more research attention has been directed towards wireless sensor networks. Once deployed, sensors are expected to operate for extended periods of time, and it is impractical to physically reach all sensors. However, it is quite often necessary to update the software running on those sensors or add new functionality to the sensors [1–3]. Reprogramming the network needs to reliably disseminate large data objects (50–100 KB) to every sensor in the network with energy efficiency [2].

Protocols for reliably disseminating large data objects in WSNs have been developed over years. Protocols in [1–4] achieve data dissemination reliability through different mechanisms such as hop-by-hop recovery, NACKs or ACKs mechanisms, while another requirement of disseminating large objects in WSNs, energy efficiency, has not been well studied.

In WSNs, energy consumption is a critical issue and sleep scheduling has been well studied as a conservative approach

to minimize the energy consumption due to *idle listening* [5, 6]. Though sleep scheduling can save energy, sensors in sleep mode cannot receive data packets. In addition, due to the unreliability of wireless communication, a sensor may not receive the packet successfully even when it is in active mode [7]. Hence, a data packet may be transmitted several times in order to be disseminated to all sensors, which wastes energy and increases the delay of the whole data dissemination process. In other words, the data dissemination process consists of sending native data packets and recovering “wanted” packets that each sensor has not received due to sleep scheduling and/or link unreliability. In order to complete the data dissemination process in a timely manner and achieve energy efficiency, it is crucial to assure that the maximum number of “wanted” packets at all sensors can be recovered at each time slot.

Recently, network coding has become a promising approach to improve the system throughput in wireless networks. Network coding with XORs operation in wireless broadcast has been studied in [8], which shows the advantage

of the proposed XORs coding scheme over the traditional wireless broadcast in the bandwidth efficiency through simulations and theoretical analysis. In XORs coding, a coded packet carries both the coding vector information and the encoded data. Thus, upon receiving a coded packet, the receiver knows which packets are encoded together and how to decode the packet with the available packets at the receiver. The work in [9] has proved that optimal XORs encoding decision for wireless broadcast, which decides the coding vector of each coded packet, is an NP-hard problem. Heuristic algorithms of encoding decision problem for wireless broadcast and multicast are proposed in [9, 10]. However, the proposed encoding decision approach can only be applied to the scenario where all receivers remain active during the whole time period of recovery. Such an approach can not be applied to WSNs with sleep scheduling because different sets of active sensors may be available at different time slots.

In this paper, given the sleep scheduling information at the sensors, we aim to determine an effective XORs encoding strategy such that the minimum number of transmissions is required in order for each sensor in the network to successfully receive the whole set of disseminated data packets. Thus, energy consumption can be reduced and the data dissemination process can be accomplished in a timely manner. To achieve such an objective, it is important to maximize the expected number of active sensors that can decode out one “wanted” data packet at each time slot in the recovery process, which is the focus of this paper. The contribution of the proposed work is summarized as follows.

- (i) The proposed work takes both link unreliability and sleep scheduling into consideration and proposes an XORs encoding decision algorithm to maximize the expected number of active sensors that can decode out one native packet in their “wanted” data packet sets at each time slot in the recovery process.
- (ii) We analyze the impact of each link’s packet loss probability and each sensor’s sleep probability at each time slot on the network coding gain, which is an extension of the analysis given in [8].
- (iii) We also study the effectiveness of sleep scheduling on energy saving, which is offset by the total number of active time slots consumed in the data dissemination process. A threshold is derived to decide whether the current sleep scheduling is effective on energy saving or not. The simulation results also confirm the accuracy of our analysis.

The rest of the paper is organized as follows. Related work is reviewed in Section 2. Section 3 introduces the system architecture and data dissemination schemes. The problem description and its complexity is presented in Section 4. Section 5 describes the algorithm design. Theoretical analysis is given in Section 6. Section 7 gives the simulation results. Finally, we conclude the paper in Section 8.

2. Related Work

In this section, we review the related work of network coding in WSNs. Network coding is originally proposed in information theory [11] and recently has become a promising approach to improve the system throughput in wireless networks [11–16]. Adaptive network coding is proposed in [17] to reduce traffic in the process of software updates where linear network coding technique is used. As computation ability and the memory at sensor nodes are very limited, the complexity of linear encoding and decoding introduces extra overhead. Thus, it is more appropriate to use XORs operation in WSNs since both encoding and decoding operations are much simpler. In fact, XORs coding has been widely used in wireless networks to reduce the complexity of linear network coding [8, 10, 18, 19].

COPE proposed in [18] improves the throughput of unicast with XORs coding. By exploiting the broadcast nature of wireless medium, each node buffers overheard packets for a short time and notifies its neighbors which packets it has heard. When a node transmits a packet, it uses its knowledge of what its neighbors have heard to perform *opportunistic coding* and XORs multiple packets to transmit them as a single packet while ensuring that each intended next-hop has enough information to decode the encoded packet.

Network coding with XORs operation in wireless broadcast has also been studied in [8], which shows the advantage of the proposed network coding scheme over traditional wireless broadcast in bandwidth efficiency through simulations and theoretical analysis. However, encoding decision has not been given in [8]. The work in [9] has proved that optimal XORs encoding decision problem for wireless broadcast is an NP-hard problem.

Several heuristic algorithms for encoding decision in wireless broadcast and multicast have been proposed in [9, 10]. With the knowledge of the “wanted” packet set at each receiver, an auxiliary graph is constructed. The encoding decision during the recovery process is then converted to a clique partition problem in the auxiliary graph. However, the proposed encoding decision algorithms can only be applied to the scenario where all receivers remain active during the time period of recovery. Such an approach cannot be applied to WSNs since different set of active sensors may be available at different time slots. Thus, encoding decision in WSNs with sleep scheduling cannot be converted into finding a minimum clique partition in the graph.

The work in [20] proposes a retransmission scheme, which only uses reception estimation to determine the coding set selection. However, the reception estimation at the source node may not be accurate enough, consequently, some receivers may not be able to decode useful information from the coded packet and more retransmissions will be needed. In addition, the coding decision based on reception estimation does not consider the impact of sleep scheduling, which affects the decoding probability at the receivers in low duty-cycled WSNs.

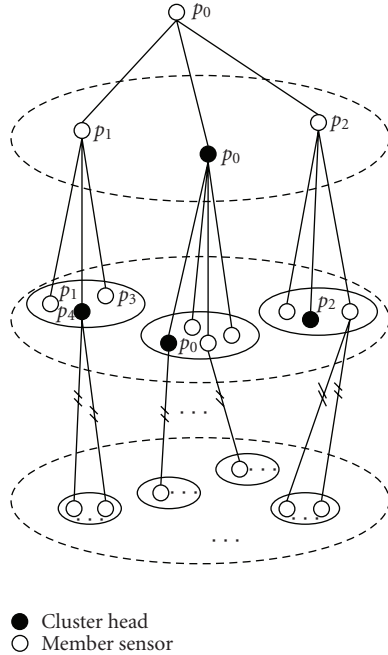


FIGURE 1: Hierarchical architecture.

In this paper, we propose to use XORs coding in data dissemination in a large scale WSN which is organized as a multihop cluster hierarchy [21]. A multihop cluster hierarchical architecture consists of multiple layers as shown in Figure 1. In the lowest layer, all the nodes in the network are grouped into clusters. In addition, besides being a member in a cluster, a node may act as a cluster head in a down layer cluster, for example, p_2 in the figure. Within each cluster, the cluster head communicates with its member sensors in a one-hop fashion [22]. We also assume that each sensor is aware of its one-hop neighbors' sleep scheduling and the reliability of the wireless links between the sensor to its neighbors. This can be easily accomplished by one-hop information exchange and link loss inference [23].

3. System Architecture and Data Dissemination with Network Coding

Our data dissemination process is conducted at each cluster head so as to make sure that finally all the sensors obtain the updating packets. In a multihop cluster hierarchy, if a cluster head in an intermediate layer starts to transmit the received packet immediately after receiving one fresh packet, the gain of network coding cannot be fully utilized. On the other hand, if a cluster head waits and starts to transmit packets until it receives all packets from the cluster head in the upper layer, it will waste bandwidth and introduce extra delay. In order to achieve the balance between bandwidth efficiency and network coding gain, we propose to use a threshold α to determine when the current cluster head starts to transmit the packets to its member nodes. Specifically, for each cluster head, after obtaining $\alpha M'$ fresh native packets, where $0 < \alpha \leq 1$ and M' is the number of native packets available

at its upper-layer cluster head, it will conduct XORs coding scheme to transmit the packets to its member nodes. In the simulation part, we will study the impact of the threshold α on the delay and energy consumption.

In the rest of the paper, we focus on how a cluster head encodes the packets and transmits them to its member sensors. The coding decision at other cluster heads can use the same approach.

As we mentioned earlier, the data dissemination process consists of sending native data packets and recovering "wanted" packets for each receiver. We now give an example to show that network coding can indeed recover "wanted" packets for all neighbors more efficiently.

Suppose that four packets d_1, d_2, d_3 , and d_4 need to be transmitted to sensors p_1, p_2, p_3 and p_4 as shown in Figure 2. The sleep scheduling at each receiver is given in Figure 2(a) where 1 denotes that this sensor is active at the current time slot, otherwise, it is in sleep mode. For the sake of simplicity, in this example, we assume that no packet is lost due to unreliable wireless communication, which means that a sensor can receive a packet successfully when it is in active mode. We also assume that an active sensor can only transmit or receive one packet at each time slot [5]. We show that different data dissemination approaches will lead to different finishing time of data dissemination.

- (i) Without network coding, 4 native packets will be sent firstly, followed by sending native packets to recover "wanted" packets at sensors. Figure 2(b) gives the "wanted" data packet set at each sensor after 4 native packets are sent out. Without network coding, it will take 10 time slots to finish the data dissemination process as shown in Figure 2(c).
- (ii) With network coding, 4 native packets can be sent at first followed by sending encoded packets to recover "wanted" packets at sensors. Assume that our coding strategy at each time slot is to maximize the number of active receivers that can decode the encoded packet. For example, at time t_5 , if $d_1 \oplus d_2$ is sent, all four receivers can obtain a "wanted" packet by $d_1 \oplus (d_1 \oplus d_2)$ or $d_2 \oplus (d_1 \oplus d_2)$. Eventually, it will take 8 time slots to finish the data dissemination process as shown in Figure 2(d). Under such a data dissemination approach, as all native packets are sent at first, the available packets at sensors are most diversified. Thus, the best network coding gain can be achieved. This, however, means that each sensor needs to buffer all received native packets in order to decode out "wanted" packets, which might not be feasible in a WSN due to limited memories at sensors.
- (iii) An alternative approach will be to divide the data dissemination process into several batches where in each batch, M native packets are sent followed by the recovering process [24]. Once all M native packets are received by all sensors in the cluster, the cluster head proceeds to transmit the following batch of packets. The data dissemination is accomplished when all batches of packets are obtained by all sensor nodes in the network. In Figure 2(e), we send two native

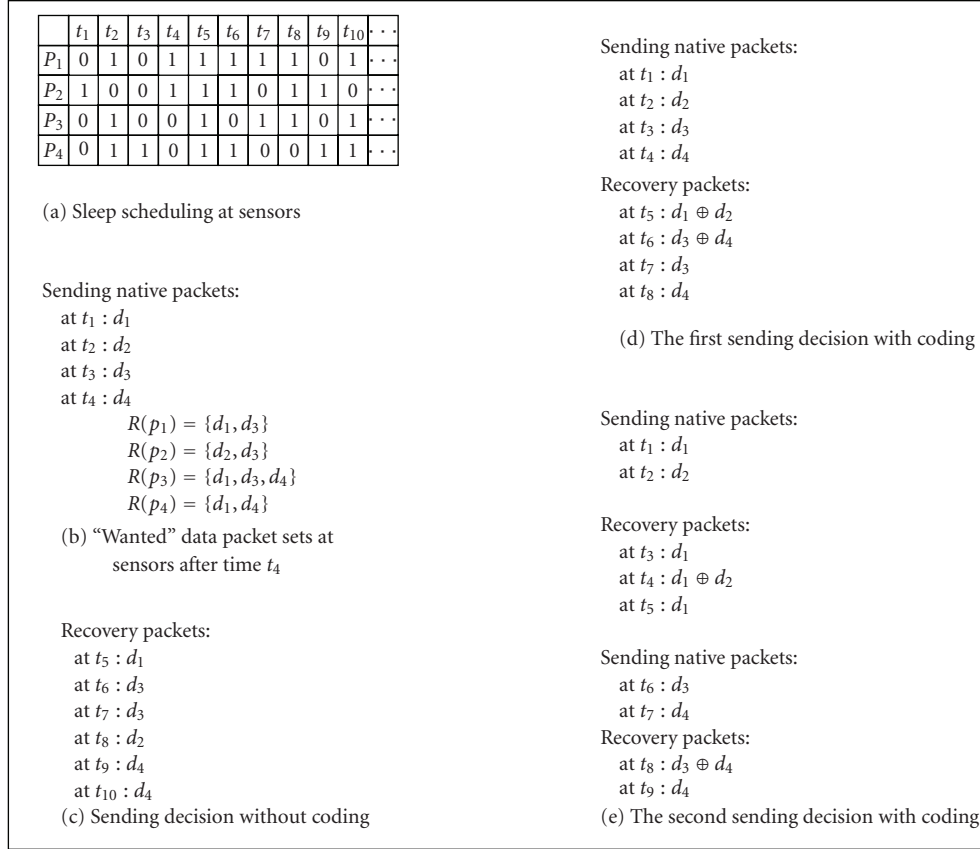


FIGURE 2: Comparison with different dissemination schemes.

packets at first, followed by sending encoded packets to recover "wanted" packets of the first batch at sensors, then send the last two native packets followed by sending encoded packets to recover "wanted" packets of the second batch at sensors. It takes 9 time slots to finish the data dissemination process.

We now discuss how the cluster head can maintain "wanted" packet set at each member sensor. After sending out a packet, the cluster head needs to collect the "wanted" packet set at each member sensor. In order to reduce ACKs implosion, only the active receivers that have received a packet at current time slot successfully and can obtain/decode one "wanted" packet from the received packet will send an ACK message to the cluster head. Thus, according to ACKs from receivers, the cluster head can derive the "wanted" packet set for each active receiver.

With the information of "wanted" packet set of each receiver at each time slot in the recovery process, an encoding decision which aims to maximize the expected number of active sensors that can decode out one "wanted" packet at current time slot will be introduced in the following section.

4. Problem Description and Complexity

In this section, we first describe the encoding decision problem that aims to decide which native packets should be

encoded at each time slot t in the recovering process such that the maximum expected number of active sensors at time slot t can decode out one "wanted" native packet. Thus, we limit our discussion to the recovery process of one data dissemination batch in a cluster, which can also be applied to other batches in all other clusters.

Suppose that $D = \{d_1, d_2, \dots, d_M\}$ is the set of data packets in a batch which need to be disseminated to all the sensors in a cluster. Let $P_t = \{p_{i_1}, p_{i_2}, \dots, p_{i_l}\}$ be the set of active member sensors in the cluster at t th time slot. At each time slot, the cluster head can obtain its neighbor sensors' "wanted" packet set based on ACKs feedback. Let $r_{i,j}$ be 1 if packet d_j is not available at active sensor p_i at current time slot where $d_j \in D$, otherwise, let it be 0. Let $R(p_i) = \{d_j \mid r_{i,j} = 1 \text{ and } p_i \in P_t\}$ be the "wanted" data packet set of active sensor p_i at current time slot t as shown in Figure 2(b). Assume that l_i is the probability that sensor p_i can not successfully receive a packet from the cluster head when p_i is in active mode.

Let a_j be 1 if native packet $d_j \in D$ is combined in current encoded packet, otherwise, let it be 0. Let $c_{i,j}$ be 1 if active sensor p_i can decode out one "wanted" native packet d_j from the current encoded packet where $d_j \in R(p_i)$, otherwise, let it be 0. Considering unreliable wireless communication, the probability that an active sensor p_i can successfully obtain one "wanted" packet at the current time slot is $\sum_{j=1}^M c_{i,j}(1-l_i)$. Thus, at current time slot, the expected

number of sensors that can decode out one “wanted” packet is $\sum_{i \in \{i | p_i \in P_t\}} \sum_{j=1}^M c_{i,j} (1 - l_i)$, which needs to be maximized in order to save energy.

Still take Figure 2(d) as an example, after t_4 , the cluster head starts to recover the “wanted” packets at its member sensors. At t_5 , if the cluster head sends an encoded packet $d_1 \oplus d_2$, in an ideal condition where no packet will be lost, active receivers p_1, p_2, p_3, p_4 can decode out one “wanted” packet by $d_1 \oplus (d_1 \oplus d_2)$ or $d_2 \oplus (d_1 \oplus d_2)$. Assume that $l_1 = 0.1$, $l_2 = 0.2$, $l_3 = 0.3$, $l_4 = 0.15$ in a practical wireless network where the probability of successfully receiving a packet at p_1, p_2, p_3 , and p_4 is 0.9, 0.8, 0.7 and 0.85 respectively due to unreliable wireless communication. Thus, the expected number of active receivers that can decode out one “wanted” packet after receiving the current encoded packet $d_1 \oplus d_2$ is $0.9 * 1 + 0.8 * 1 + 0.7 * 1 + 0.85 * 1 = 3.25$, which is maximum at the current time slot. Thus, the cluster head will send out $d_1 \oplus d_2$ at the current time slot. In this paper, such an encoding decision problem using XORs coding is referred to as *network coding based data dissemination* (NCDD) problem.

4.1. Problem Formulation. We can formally formulate the NCDD problem at time slot t in the recovery process as follows:

$$Z = \max \sum_{p_i \in P_t} \sum_{j=1}^M c_{i,j} * (1 - l_i) \quad (1)$$

subject to

$$\sum_{j=1}^M c_{i,j} \leq 1, \quad \forall i \in \{i' | p_{i'} \in P_t\}. \quad (2)$$

$$c_{i,j} \leq a_j * r_{i,j}, \quad \forall i \in \{i' | p_{i'} \in P_t\}, \forall 1 \leq j \leq M, \quad (3)$$

$$a_{j'} * r_{i,j'} + c_{i,j} \leq 1, \quad i \in \{i' | p_{i'} \in P_t\}, \forall 1 \leq j \neq j' \leq M, \quad (4)$$

$$a_j \leq \sum_{i \in \{i' | p_{i'} \in P_t\}} r_{i,j}, \quad \forall 1 \leq j \leq M, \quad (5)$$

$$a_j, c_{i,j} \in \{0, 1\}, \quad \forall i \in \{i' | p_{i'} \in P_t\}, \forall 1 \leq j \leq M. \quad (6)$$

In the above formulation, the term of the objective represents the expected number of active receivers that can decode out one “wanted” data packet from the encoded packet at the current time slot. Equations (2) and (6) ensure that each receiver can only decode out at most one “wanted” native packet from the encoded packet. Equations (3) and (4) give two requirements that active receiver p_i can decode out one “wanted” packet d_j : (1) packet d_j is in p_i 's “wanted” packet set and d_j is participated in the encoded packet; (2) all other combined native packets except d_j in the encoded packet have already been successfully received by receiver p_i . Equation (5) guarantees that if packet d_j is available at all active receivers at current time slot t , d_j must not be combined into the encoded packet.

4.2. Problem Complexity

Theorem 1. *NCDD problem is NP-hard.*

Proof. We prove the theorem by a reduction from MAXIMUM ONE-IN-THREE SAT problem which is a well known NP-hard problem in the strong sense.

MAXIMUM ONE-IN-THREE SAT: We are given a set $U = \{u_1, u_2, \dots, u_M\}$ of M boolean variables and a collection $C = \{c_1, c_2, \dots, c_n\}$ of clauses with exactly three literals. Each of these clauses is a boolean formula and it is true if and only if exactly one of its three literals is true. Without loss of generality, we assume that the three literals in c_i are $\{u_{i_1}, u_{i_2}, u_{i_3}\}$. The objective of MAXIMUM ONE-IN-THREE SAT is to find a truth assignment such that the maximum number of clauses is true. We use OPT^s to denote the optimal solution of this problem.

Given an instance of MAXIMUM ONE-IN-THREE SAT, we can construct an instance of the decision version of the NCDD problem in polynomial time as follows. Let there be M data packets needed to be disseminated from the cluster head to n receiver nodes. If $u_j = 1$, packet d_j is participated in encoding, otherwise, d_j is not participated in encoding. For each clause c_i , if u_j is a literal of c_i , then d_j is a “wanted” packet at p_i . In other words, each sensor p_i has lost exactly three packets and has all other packets. Let the probability that an active sensor can successfully receive a packet be 100%. Then, our objective is to maximize $\sum_{i \in \{i | p_i \in P_t\}} \sum_{j=1}^M c_{i,j}$ For a given encoded packet, p_i can decode a new native packet if and only if exactly one native packet in R_i is encoded into the new encoded one. The problem is to find an encoding strategy to maximize the number of receivers which can decode out one “wanted” packet from the encoded packet. We use OPT^p to refer to the result of this objective.

- (i) Suppose that there is a true assignment for MAXIMUM ONE-IN-THREE SAT with the maximum number of clauses. If c_i is true, there must be exactly one true assignment for $\{u_{i_1}, u_{i_2}, u_{i_3}\}$. Without loss of generality, we assume that u_{i_2} is true while u_{i_1}, u_{i_3} are both false. According to the construction of the instance, only d_{i_2} is participated in encoding while neither d_{i_1} nor d_{i_3} is participated in encoding. In other words, only one lost packet of p_i is participated in encoding and p_i has all other packets involved in encoding, thus, p_i can decode out one “wanted” native packet d_{i_2} . Therefore, if there is a clause which is true in the MAXIMUM ONE-IN-THREE SAT problem, there must be a receiver which can obtain a “wanted” native packet. Then, we have $OPT^s \leq OPT^p$.
- (ii) Suppose that there is an encoding strategy such that the maximum number of receivers can decode the new native packet. Assume that p_i can decode a new native packet d_{i_2} from the encoded one. According to the decoding strategy, the other two “wanted” packets d_{i_1}, d_{i_3} must not be encoded into the new one, that is, u_{i_1}, u_{i_3} both have false assignment while u_{i_2} is true.

In this assignment, c_i also has a true value. So, we have $OPT^P \leq OPT^S$.

The above analysis shows that $OPT^P = OPT^S$. Thus NCDD problem is NP-hard. \square

5. Algorithm for NCDD Problem

In this section, we first introduce an auxiliary graph in which each vertex is assigned a weight. We then show that the proposed NCDD problem can be converted into finding a maximum weight clique problem in the auxiliary graph, based on which we develop a heuristic algorithm for the NCDD problem.

5.1. Model Design. At any t th time slot, let $R(p_i) \subseteq D$ be the set of packets “wanted” by p_i and $H(p_i) \subseteq D$ be the set of packets received by p_i . We can construct an auxiliary graph $G(V, E)$ similar to [9] where $V = \{v_{i,j} \mid d_j \in R(p_i) \text{ and } p_i \in P_t\}$, which means that every “wanted” packet of each active sensor has a vertex in G . Considering two receivers p_{i_1} and p_{i_2} , if they have lost the same packet d_j , then they can both recover d_j if only native packet d_j is encoded at current time slot. We use a link $e \in E$ between $v_{i_1,j}$ and $v_{i_2,j}$ to denote such recoverability. If d_{j_1} is a “wanted” packet of p_{i_1} and $d_{j_1} \in H(p_{i_2})$, while d_{j_2} is a “wanted” packet of p_{i_2} and $d_{j_2} \in H(p_{i_1})$, then p_{i_1} can recover d_{j_1} when it receives $d_{j_1} \oplus d_{j_2}$ and p_{i_2} can recover d_{j_2} when it receives $d_{j_1} \oplus d_{j_2}$. We use a link $e \in E$ between v_{i_1,j_1} and v_{i_2,j_2} to denote such recoverability. In other words, $E = \{(v_{i_1,j_1}, v_{i_2,j_2}) \mid d_{j_2} \in H(p_{i_1}), d_{j_1} \in H(p_{i_2}) \text{ or } j_1 = j_2, i_1 \neq i_2\}$ where $p_{i_1}, p_{i_2} \in P_t$.

For a clique $Q = \{v_{i_1,j_1}, v_{i_2,j_2}, \dots, v_{i_k,j_k}\}$ in the graph, let $P' = \{p_i \mid v_{i,j} \in Q, 1 \leq j \leq M\}$ be the sensors which have “wanted” packets in Q and $D' = \{d_j \mid v_{i,j} \in Q, 1 \leq i \leq n\}$ be the set of “wanted” packets of those sensors in Q . Suppose that there are m' packets in D' . For any vertex $v_{i,j} \in Q$, according to the edge assignment of G , p_i must have already successfully obtained the packets in $D' - \{v_j\}$ but still requires packet v_j . Thus, if $d_{j_1} \oplus d_{j_2} \oplus \dots \oplus d_{j_{m'}}$ where $d_{j_1}, d_{j_2}, \dots, d_{j_{m'}} \in D'$ are encoded and sent at t th time slot, each sensor in P' will be able to decode out one “wanted” packet if the encoded packet can be successfully received by all sensors in P' . To consider the unreliability of wireless communication, we assign weight $w_{i,j} = 1 - l_i$ in the vertex $v_{i,j}$ for any $j \in \{j \mid v_{i,j} \in V\}$. Then the weight for clique Q which is defined in

$$w(Q) = \sum_{(i,j) \in \{(i,j) \mid v_{i,j} \in Q\}} w_{i,j}, \quad (7)$$

is equivalent to the expected number of active sensors which can successfully decode out one “wanted” packet if all packets in D' are encoded together. Thus, our NCDD problem which aims to maximize the expected number of active sensors that can decode out one “wanted” packet is converted into finding a maximum weight clique in graph G .

For example, after the whole 4 native packets are sent, the “wanted” packet set in Figure 2(b) can be constructed into Figure 3. Thus, the encoding decision for recovery process at

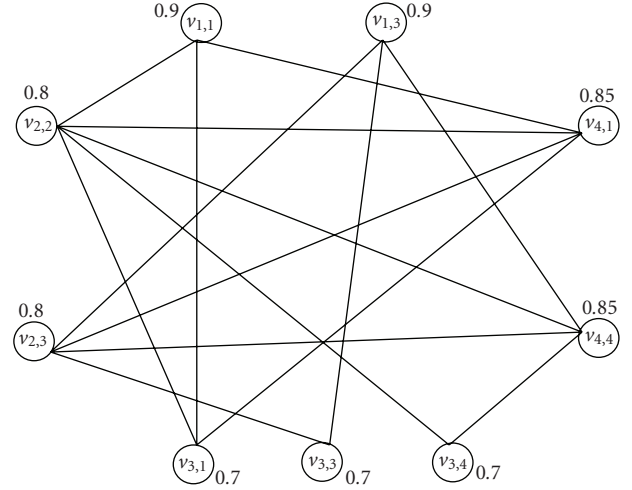


FIGURE 3: Graph model.

t_5 is then converted into finding a maximum weight clique in such a graph. As shown in Figure 3, the clique that consists of $\{v_{1,1}, v_{2,2}, v_{3,1}, v_{4,1}\}$ is the clique with the maximum weight $0.9 + 0.8 + 0.7 + 0.85 = 3.25$. After the encoded packet $d_1 \oplus d_2$ is sent, active receivers p_1, p_2, p_3, p_4 can decode out d_1, d_2, d_1, d_1 , respectively, if all sensors successfully receive $d_1 \oplus d_2$.

5.2. Algorithm Design. Assume that the total number of vertices in $G(V, E)$ is N . We first sort all vertices into nonincreasing order according to $w_{i,j}$. For the example given in Figure 3, vertices in G will be sorted into $V = \{v_{1,1}, v_{1,3}, v_{4,1}, v_{4,4}, v_{2,2}, v_{2,3}, v_{3,1}, v_{3,3}, v_{3,4}\}$.

For the simplicity of presentation, we abuse the notation a little bit and assign a unique id v_k for each vertex in G , which uses one-dimensional subscript for vertices in G instead of using two-dimensional subscripts. Correspondingly, we use w_k to denote the weight of v_k . Thus, for the example given in Figure 3, we have $V = \{v_1(v_{1,1}), v_2(v_{1,3}), v_3(v_{4,1}), v_4(v_{4,4}), v_5(v_{2,2}), v_6(v_{2,3}), v_7(v_{3,1}), v_8(v_{3,3}), v_9(v_{3,4})\}$. Without loss of generality, we assume that $V = \{v_1, v_2, \dots, v_N\}$ where $w_1 \geq w_2 \geq \dots \geq w_N$.

Let Q_i be the clique with maximum weight in the subgraph which only contains vertices of $S_i = \{v_i, v_{i+1}, \dots, v_N\}$ and let $C(Q_i)$ be the weight of clique Q_i . In other words, Q_i represents the maximum weight clique the algorithm has found considering of the subgraph consisting of vertices $\{v_i, v_{i+1}, \dots, v_N\}$. The algorithm starts with $i = N$ and iteratively considers more vertices until all vertices in G are considered. The algorithm stops when Q_1 is found.

When we consider vertex v_{i-1} , there are two cases. If $Q_i \cup \{v_{i-1}\}$ is also a clique, then $Q_{i-1} = Q_i \cup \{v_{i-1}\}$ and $C(Q_{i-1}) = C(Q_i) + w_{i-1}$, otherwise, if $Q_i \cup \{v_{i-1}\}$ is not a clique, we need to find out a clique Q_{i-1} that includes v_{i-1} in the subgraph consisting of $S_{i-1} = \{v_{i-1}, v_i, \dots, v_N\}$. Let $N(v_{i-1})$ be the set of neighbors of vertex v_{i-1} . Initially, $Q_{i-1} = \{v_{i-1}\}$ and $S_{i-1} = N(v_{i-1}) \cap S_{i-1}$. If S_{i-1} is not \emptyset , let j be the smallest j such that $v_j \in S_{i-1}$. We add v_j to the clique, that is, $Q_{i-1} = Q_{i-1} \cup \{v_j\}$,

```

Function wclique( $S_i, C(Q_i), i$ )
    if  $|S_i| = \emptyset$ 
        if  $C(Q_i) > \max$ 
             $\max = C(Q_i)$ ;
             $mc = i$ ;
        return
    while  $S_i \neq \emptyset$ 
         $j = \min\{j \mid v_j \in S_i\}$ ;
         $Q_i = Q_i \cup \{v_j\}$ ;
         $C(Q_i) = C(Q_i) + w_j$ ;
         $S_i = (S_i - \{v_j\}) \cap N(v_j)$ ;
        if  $S_i = \emptyset$ 
            if  $C(Q_i) > \max$ 
                 $\max = C(Q_i)$ ;  $mc = i$ ;
        return
Function MWC
     $Q_i = \emptyset, i \in \{1, \dots, N-1\}$ ;  $Q_N = \{v_N\}$ ;
     $C(Q_i) = 0, i \in \{1, 2, \dots, N-1\}$ ;
     $C(Q_N) = w_N$ ;
     $\max = C(Q_N)$ ;
     $S_i = \{v_i, v_{i+1}, \dots, v_N\}, i \in \{1, 2, \dots, N\}$ ;
    for  $i = N-1$  down to 1
        if  $\{v_i\} \cup Q_{i+1}$  is also a clique;
             $Q_i = Q_{i+1} \cup \{v_i\}$ ;
             $C(Q_i) = C(Q_{i+1}) + w_i$ ;
        else
             $Q_i = \{v_i\}$ ;
             $C(Q_i) = w_i$ ;
            wclique( $S_i \cap N(v_i), w_i, i$ );
        if  $mc \neq i$ 
             $Q_i = Q_{i+1}$ ;  $C(Q_i) = C(Q_{i+1})$ ;
    return
    
```

ALGORITHM 1: Maximum weight clique algorithm.

and update S_{i-1} , that is, $S_{i-1} = S_{i-1} \cap N(v_j)$. If S_{i-1} is still not \emptyset , we then add another vertex whose index is the smallest in S_{i-1} into the clique Q_{i-1} . We repeat this process until there is no vertex in S_{i-1} , that is, $S_{i-1} = \emptyset$. By comparing the weight of the clique Q_i without including v_{i-1} and the weight of the clique Q_{i-1} including v_{i-1} , the clique Q_{i-1} with maximum weight in the subgraph including vertices in $\{v_{i-1}, v_i, \dots, v_N\}$ is set to be the one with the larger weight. The detail of the algorithm is given in Algorithm 1. After this algorithm, Q_1 gives all vertices in the found maximum weight clique. All native packets involved in Q_1 will be encoded together and be sent out at current time slot.

We now show how to find the maximum weight clique of the graph shown in Figure 3. Assume that Q_2 has been found, which consists of $\{v_2, v_4, v_6\}$. Next, we will consider Q_1 . Since $\{v_1\} \cup Q_2$ is not a clique, we need to find Q_1 which includes vertex v_1 in the subgraph consisting of $S_1 = \{v_1, v_2, \dots, v_9\}$. The corresponding steps for finding such Q_1 is given in Algorithm 2 where $v_k(v_{i_1, j_1})$ in V denotes that we use a unique id v_k in the algorithm to replace the original vertex v_{i_1, j_1} . After Q_1 is found, we compare it with Q_2 which has the weight $C(Q_2) = 2.55$. Since $C(Q_1)$ is larger than $C(Q_2)$, the clique $Q_1 = \{v_1, v_3, v_5, v_7\}$ is the maximum weight

clique found in graph G . Vertices in Q_1 indicate that p_1, p_3 and p_4 lost packet d_1 and p_2 lost packet d_2 . The encoding decision will be to send $d_1 \oplus d_2$.

6. Analysis

In this section, we firstly analyze the impact of packet loss probability and sleep probability on network coding gain. Then, we derive a threshold to decide whether the current sleep scheduling can save energy compared with no sleep scheduling. We only limit the analysis to one cluster in the multihop cluster hierarchy.

6.1. Impact of Packet Loss Probability and Sleep Probability on Network Coding Gain. Suppose that N_a is the number of transmissions that the data dissemination process requires without coding and N_b is the number of transmissions required with XORs coding. Assume that the probability that receiver p_i is in sleep mode is s_i at each time slot, and l_i is the probability that receiver p_i can not successfully receive a packet even when it is in active mode due to unreliable wireless communication. We have the following two lemmas.

Lemma 1. *The total number of transmissions without coding required for transmitting sufficient large M packets to n receivers is*

$$N_a = \sum_{i_1, i_2, \dots, i_n} \frac{(-1)^{i_1 + i_2 + \dots + i_n - 1}}{1 - (l'_1)^{i_1} (l'_2)^{i_2} \dots (l'_n)^{i_n}} M, \quad (8)$$

where $i_1, i_2, \dots, i_n \in \{0, 1\}$ and $\exists i_j \neq 0, l'_i = 1 - (1 - l_i)(1 - s_i)$.

Proof. See Appendix A. \square

Lemma 2. *The total number of transmissions with XORs coding for transmitting sufficient large M packets to n receivers is*

$$N_b = \frac{M}{\min_{i \in \{1, 2, \dots, n\}} \{(1 - l_i)(1 - s_i)\}}. \quad (9)$$

Proof. See Appendix B. \square

With the analytical result of N_a and N_b , we can define analytical network coding gain as

$$\gamma = \frac{N_a - N_b}{N_a}. \quad (10)$$

Take two receivers as an example, assume that $l_1 = 0.1, l_2 = 0.25, s_1 = 0.15, s_2 = 0.05$ and M is sufficient large. According to (8) and (9), we can calculate that $N_a = 1.6382M, N_b = 1.4035M$. Then, the analytical network coding gain is $\gamma = 0.1433$.

From Lemmas 1 and 2, we can also obtain the following corollary.

Corollary 1. *With two receivers, the maximum network coding gain γ can be achieved if $l'_1 = l'_2$, that is, $(1 - l_1)(1 - s_1) = (1 - l_2)(1 - s_2)$.*

Proof. See Appendix C. \square

$V = \{v_1(v_{1,1}), v_2(v_{1,3}), v_3(v_{4,1}), v_4(v_{4,4}), v_5(v_{2,2}), v_6(v_{2,3}), v_7(v_{3,1}), v_8(v_{3,3}), v_9(v_{3,4})\}$	
Step 1:	$Q_1 = \{v_1\}, S_1 = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}, N(v_1) = \{v_4, v_5, v_7\} \quad C(Q_1) = 0.9$
	$S_1 = S_1 \cap N(v_1) = \{v_3, v_5, v_7\} \quad v_j = v_3$
	$Q_1 = Q_1 \cup \{v_3\} = \{v_1, v_3\} \quad C(Q_1) = 0.9 + 0.85 = 1.75$
Step 2:	$N(v_3) = \{v_1, v_5, v_6, v_7\}$
	$S_1 = S_1 \cap N(v_3) = \{v_5, v_7\} \quad v_j = v_5$
	$Q_1 = Q_1 \cup \{v_5\} = \{v_1, v_3, v_5\} \quad C(Q_1) = 1.75 + 0.8 = 2.55$
Step 3:	$N(v_5) = \{v_1, v_3, v_4, v_7, v_9\}$
	$S_1 = S_1 \cap N(v_5) = \{v_7\} \quad v_j = v_7$
	$Q_1 = Q_1 \cup \{v_7\} = \{v_1, v_3, v_5, v_7\} \quad C(Q_1) = 2.55 + 0.7 = 3.25$
Step 4:	$N(v_7) = \{v_1, v_3, v_5\}$
	$S_1 = S_1 \cap N(v_7) = \Phi$
	Terminate;

ALGORITHM 2: The steps of finding Q_1 .

6.2. *Impact of Sleep Probability on Energy Consumption.* Though sleep scheduling can save energy consumption due to *idle listening*, sensors in sleep mode cannot receive data packets, which imposes retransmission and may consume more energy. If sensor p_i is active at t th time slot, we say that t th time slot is an active time slot for sensor p_i . We know that only at its active time slot, sensor p_i consumes its energy. Thus, we can use the total number of active time slots consumed for the sensors to successfully receive the whole set of packets as the energy consumption for data dissemination.

We define a threshold as follows:

$$\varepsilon = \sum_{i=1}^n (1 - s_i) - \frac{n}{l_{\min}} \min_{i \in \{1, 2, \dots, n\}} \{(1 - l_i)(1 - s_i)\}, \quad (11)$$

where $l_{\min} = 1 - \max_{i \in \{1, 2, \dots, n\}} \{l_i\}$.

Then, we have the following lemma.

Lemma 3. *In XORs coding, if $\varepsilon < 0$, the current sleep scheduling can save energy consumed by idle listening; otherwise, the current sleep scheduling has no contribution to energy saving.*

Proof. See Appendix D. \square

Take two receivers with $l_1 = 0.23$, $s_1 = 0.15$, $l_2 = 0.27$, $s_2 = 0.18$ as an example, according to (11), we have $\varepsilon > 0$. Thus, the energy saving with sleep scheduling is offsetted by more retransmissions. In this case, the cluster head should wake up more sensors. An interesting problem is how to design an optimal sleep scheduling such that energy saving of sleep scheduling will not be offsetted by more retransmission, which is out of the scope of this paper.

7. Simulation Results

In this section, we demonstrate the effectiveness of our dissemination schemes through simulations using C++ simulator. In our simulations, a multihop cluster hierarchical

WSN is randomly generated with the fixed value of the number of sensors if without specification. We group the packets required to send into batches, and each batch has M packets. Recovery process with network coding starts after every M native packets are transmitted. In a cluster, we randomly generate sensor p_i 's sleep scheduling according to its sleeping probability s_i .

To demonstrate the advantage of our coding scheme, we introduce two baseline algorithms, namely, *dissemination without coding* algorithm and *dissemination with random coding* algorithm. *Dissemination without coding* algorithm randomly transmits a native "wanted" packet at each time slot until all receivers obtain their "wanted" data packets while *dissemination with random coding* algorithm transmits an XORs packet which is randomly generated at each time slot until all receivers obtain their "wanted" packets.

In the simulation, we are interested in evaluating the performance of our coding schemes from the following perspectives.

- (i) The number of active receivers that can obtain a new "wanted" packet at one time slot and the total number of transmissions required in one batch data dissemination within one cluster.
- (ii) The impact of the number of receiver sensors n , batch size M , sleep probability and packet loss probability on the network coding gain under different dissemination schemes within one cluster.
- (iii) How close the performance of our proposed algorithms is to the derived analytical results within one cluster.
- (iv) The impact of the threshold α on the delay and the total number of transmissions required in a multihop cluster hierarchy.

For each setting, we simulate 150 instances and report the average performance.

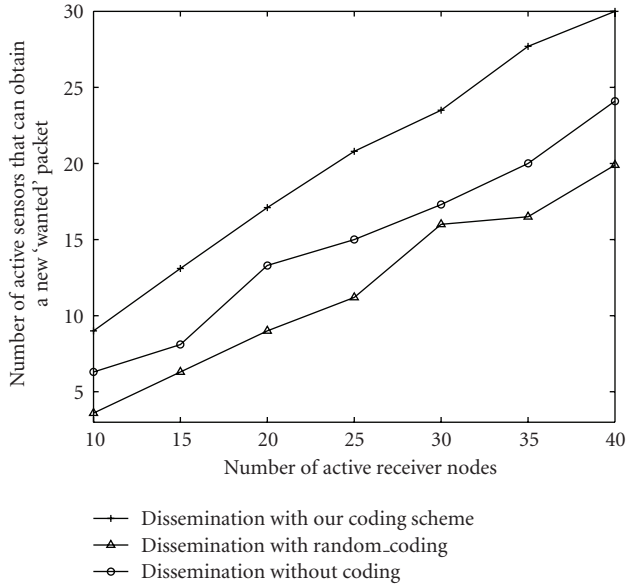


FIGURE 4: The number of receivers that can obtain a new “wanted” packet at a time slot versus the number of receiver nodes.

7.1. Comparison with Different Data Dissemination Schemes. The effectiveness of our coding scheme for maximizing the expected number of sensors that can obtain one “wanted” packet at one time slot is demonstrated by comparing with *dissemination without coding* algorithm and *dissemination with random_coding* algorithm.

We evaluate the performance of our algorithms by varying the number of active sensors within a cluster at one time slot in the range of [10, 40] for $M = 50$, and $l_i = 0.2$. As shown in Figure 4, the number of active sensors that can obtain one “wanted” packet by our coding scheme is much more than that by *dissemination without coding* algorithm and *dissemination with random_coding* algorithm.

For one batch data dissemination process within a cluster, to demonstrate the performance of our coding scheme, the total number of transmissions required is also compared with the other two baseline algorithms: *dissemination without coding* and *dissemination with random_coding* algorithms. We vary the number of packets needed to be sent in the range of [60, 100] for $n = 10$, $s_i = 0.3$, $l_i = 0.2$. As shown in Figure 5, the total number of transmissions required in one batch dissemination by our coding scheme is much less than that by *dissemination without coding* and *dissemination with random_coding* algorithms. Hence, for data dissemination with a large set of packets, our XORs coding scheme can efficiently decrease the number of transmissions required. Thus, more energy can be saved.

7.2. Network Coding Gain Comparison with Analytical Results. We demonstrate the effectiveness of the proposed network coding algorithm by comparing the network coding gain obtained through simulation with the analytical network coding gain.

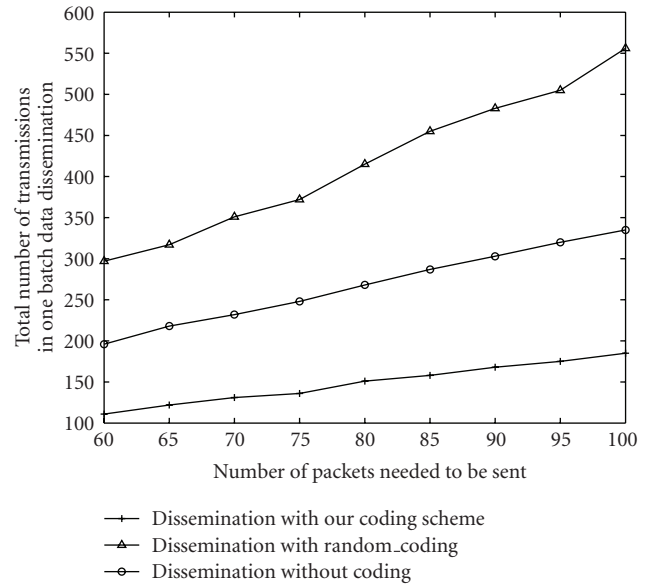


FIGURE 5: Total number of transmissions versus number of packets needed to be sent.

We start with a simple experiment where there are only two members sensors in a cluster. We fix l_1 to 0.2 and vary l_2 in the range of [0.1, 0.4] for $n = 2, s_i = 0.3, M = 100$. As shown in Figure 6(a), the network coding gain obtained by our simulation follows the same trend as the analytical results. In addition, the maximum network coding gain is achieved when $l'_1 = l'_2 = 1 - (1 - 0.2)(1 - 0.3) = 0.44$ with both our simulation results and analytical results, which verifies Corollary 1. When $l'_1 = l'_2$, most likely the “wanted” packets at one receiver are the packets available at another receiver, thus, coding opportunity is high, which achieves maximum network coding gain.

We also extend the simulation to 10 receivers in a cluster. The loss probability of p_1 is varied along the x-axis for $M = 100; s_i = 0.2, 1 \leq i \leq 5; s_i = 0.3, 6 \leq i \leq 10$ and $l_2 = l_3 = l_1 + 0.02, l_4 = l_5 = l_1 + 0.04, l_6 = l_7 = l_1 + 0.06, l_8 = l_1 + 0.08, l_9 = l_{10} = l_1 + 0.1$. As shown in Figure 6(b), the simulation results are very close to the analytical results. In addition, Figure 6 verifies that network coding indeed can bring gains on reducing the number of transmissions required.

In Figure 7, we vary the sleep probability at sensors, similar results as Figure 6 can be observed and the network coding gain obtained through simulations is quite close to the analytical results.

7.3. The Impact of Sleep Scheduling on Energy Saving. We now study the impact of sleep scheduling on the energy consumption. Our simulation is conducted within one cluster. We use the total number of active time slots consumed to denote the energy consumption in data dissemination process.

Suppose that XORs coding is applied. Let η_s be the total number of active time slots consumed for data dissemination with sleep scheduling and η_{ns} be the total number of transmissions for data dissemination without sleep scheduling.

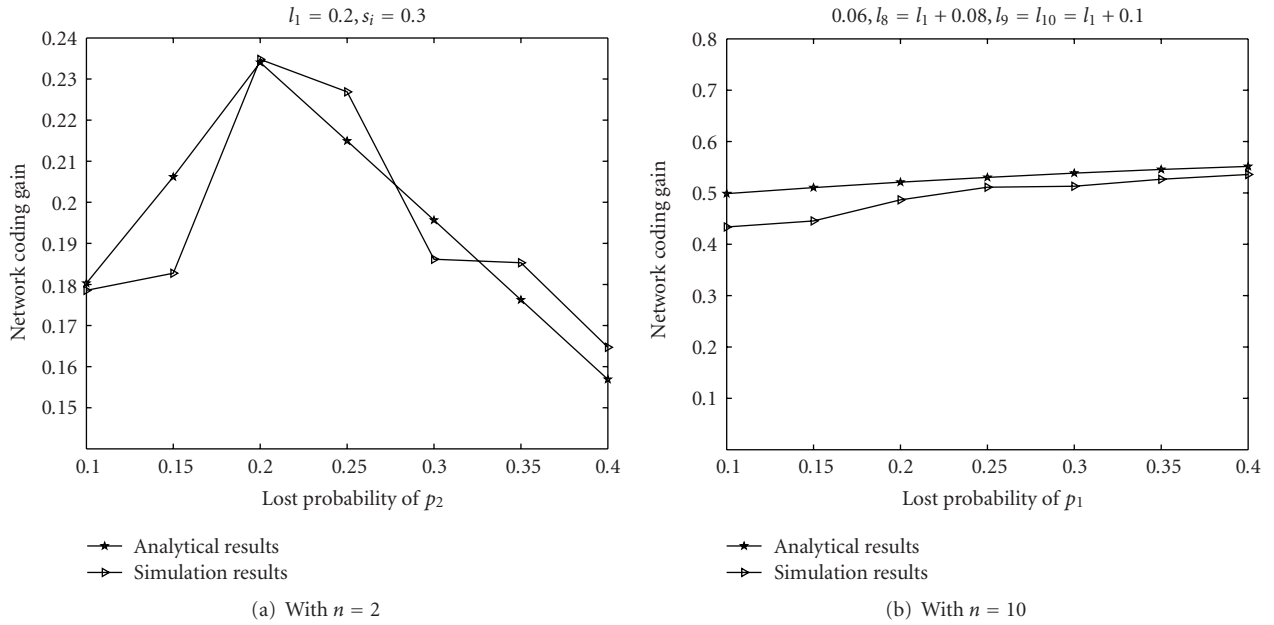


FIGURE 6: Network coding gain versus packet loss probability of sensor.

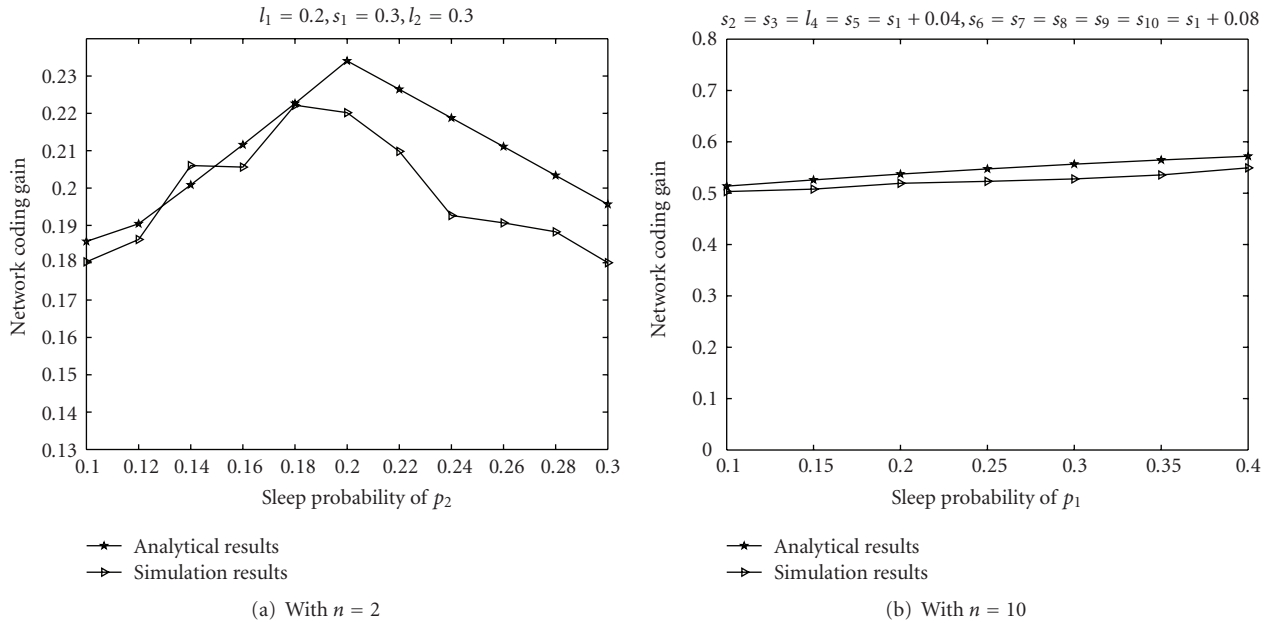


FIGURE 7: Network coding gain versus sleep probability of sensor.

The energy saving in XORs coding with sleep scheduling over that without sleep scheduling is

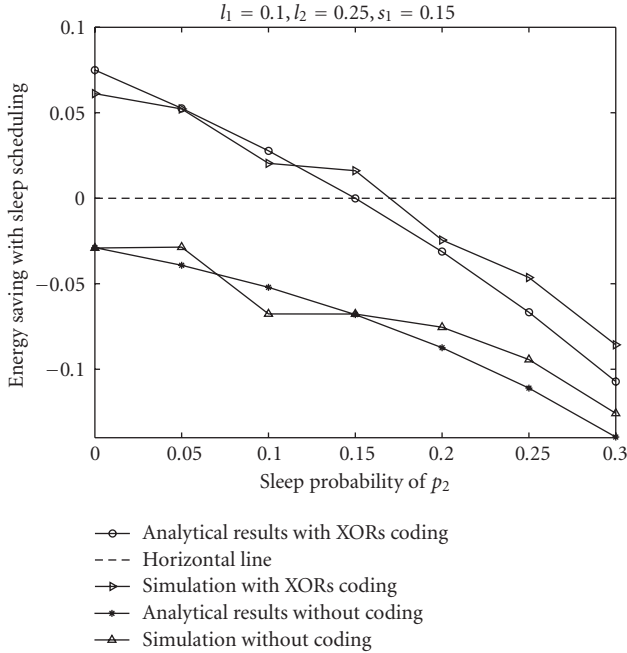
$$\delta = \frac{\eta_{ns} - \eta_s}{\eta_{ns}}. \quad (12)$$

For data dissemination without coding, we can define energy saving with sleep scheduling over that without sleep scheduling in a similar way.

We evaluate the performance of our algorithm by varying s_2 in $[0, 0.3]$ for $n = 2, l_1 = 0.1, l_2 = 0.25, s_1 = 0.15, M = 50$.

As shown in Figure 8, the simulation results are very close to the analytical results.

For our XORs coding, from the figure, we know that the energy consumption with sleep scheduling is less than that without sleep scheduling when s_2 is less than 0.15. When $s_2 = 0.15$, the energy consumption with sleep scheduling is equal to that without sleep scheduling. When s_2 is larger than 0.15, sleep scheduling has no contribution to the energy saving, it even incurs more energy consumption than that without sleep scheduling. This interesting result is plausible


 FIGURE 8: Energy saving versus sleep probability of sensor p_2 .

since when the number of sleep sensors becomes larger, more retransmissions are required, which imposes more energy consumption. In this case, the energy saving with sleep scheduling is offsetted by more retransmissions, which means that the threshold $\varepsilon > 0$ and the cluster head should wake up more sensors to receive packets in order to save energy.

7.4. The Impact of Threshold α on the Delay and the Total Number of Transmissions Required. We now study the impact of threshold α on the delay of the data dissemination process in a multihop cluster hierarchical WSN. The threshold α is varied in the range of $[0.2, 1.0]$ for $M = 30, 40, 50$. Figure 9 gives the delay required for data dissemination when the number of layers is 5 and 6, respectively. We can see that the delay increases with the threshold α . This is because the cluster heads need to wait more time before they can transmit their available packets to their members with the increasing of α . Thus, the cluster heads in down layers can do nothing for a long time. Specifically, when $\alpha = 1$, each cluster head cannot transmit its available packets until receiving all M packets. In this case, concurrent transmissions cannot be allowed even if there is no collision between them, which thus increases the delay. From Figure 9, we can also see that the delay increases with the number of layers, because the number of receivers increases with the number of layers.

We further study the impact of the threshold α on the total number of transmissions required under a multihop cluster hierarchical WSN. The threshold α is also varied in the range of $[0.2, 1.0]$ for $M = 30, 40, 50$. As shown in Figure 10, the total number of transmissions required decreases with the threshold α . When α is small, the cluster heads transmit the packets to their members more quickly. Therefore, the

number of fresh packets available at cluster heads is small, which can not fully utilize the network coding gain. Hence, the total number of transmissions required is more than with larger threshold α .

8. Conclusion

This paper studies data dissemination in wireless sensor networks with network coding to achieve energy efficiency. In order to quickly complete the whole process of data dissemination, at each time slot in the recovery process, we aim to transmit an encoded packet such that the expected number of active sensors that can decode out one “wanted” packet is maximized. A maximum weight clique model is proposed here to achieve such an objective. We further study the impact of packet loss probability and sleep probability on network coding gain. We also analyze the impact of sleep probability on energy saving gain and derive a threshold which can be used to decide whether the current sleep scheduling is effective on energy saving or not. The simulation results verify the work proposed in the paper.

Appendices

A. Proof of Lemma 1

According to [8], we can obtain that the total number of transmissions without coding to successfully deliver sufficient large M packets to n receivers is $\sum_{i_1, i_2, \dots, i_n} ((-1)^{i_1+i_2+\dots+i_n-1} / (1 - (l_1)^{i_1} (l_2)^{i_2} \dots (l_n)^{i_n})) M$, where each receiver keeps in active mode during data transmission process.

However, in the data dissemination process, receiver sensor p_i may be in sleep mode and can not successfully receive a packet. Therefore, the probability that sensor p_i can successfully receive the packet at any time slot is $(1 - l_i)(1 - s_i)$. In other words, the probability that sensor p_i will lose the packet is $1 - (1 - l_i)(1 - s_i)$. Thus, considering sleep scheduling, the total number of transmissions required without coding is

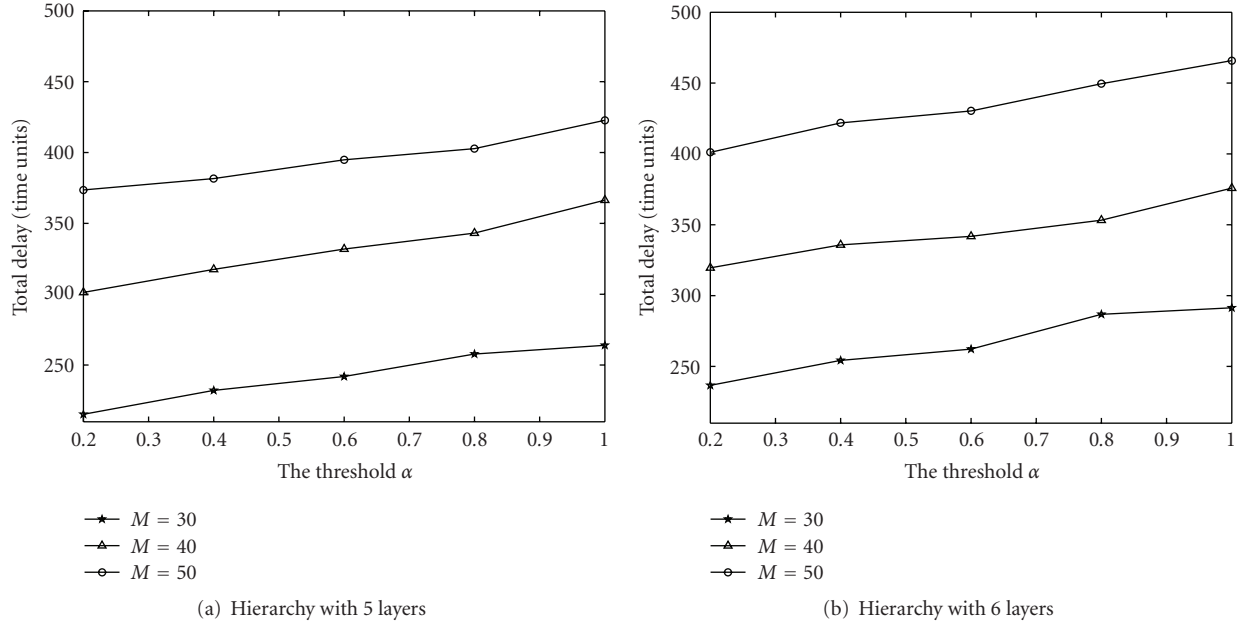
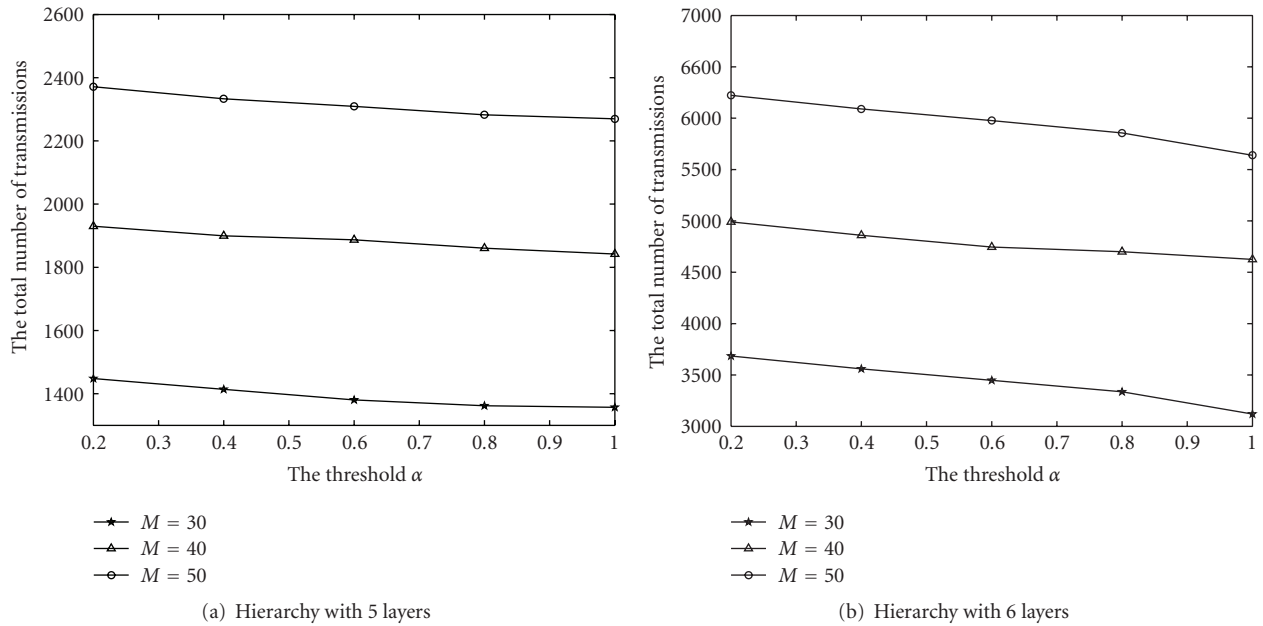
$$N_a = \sum_{i_1, i_2, \dots, i_n} \frac{(-1)^{i_1+i_2+\dots+i_n-1}}{1 - (l_1')^{i_1} (l_2')^{i_2} \dots (l_n')^{i_n}} M, \quad (\text{A.1})$$

where $i_1, i_2, \dots, i_n \in \{0, 1\}$ and $\exists i_j \neq 0, l_i' = 1 - (1 - l_i)(1 - s_i)$.

B. Proof of Lemma 2

From [8], we know that the total number of transmissions with XORs coding to successfully deliver sufficient large M packets to n receivers is $M / (1 - \max_{i \in \{1, 2, \dots, n\}} \{l_i\})$, where each receiver keeps in active mode during the data transmission process.

As in Appendix A, the probability that sensor p_i can not successfully receive the packet with sleep scheduling is changed into $1 - (1 - l_i)(1 - s_i)$. Thus, the total number

FIGURE 9: The total delay (time units) versus the threshold α .FIGURE 10: The total number of transmissions required versus the threshold α .

of transmissions required with XORs coding to transmit sufficient large M packets to n receivers is

$$\begin{aligned}
 N_b &= \frac{M}{1 - \max_{i \in \{1, 2, \dots, n\}} \{1 - (1 - l_i)(1 - s_i)\}} \\
 &= \frac{M}{\min_{i \in \{1, 2, \dots, n\}} \{(1 - l_i)(1 - s_i)\}},
 \end{aligned} \tag{B.2}$$

C. Proof of Corollary 1

With two receivers, from Lemma 1, the total number of transmissions required for M packets without coding is $N_a = M/(1 - l'_1) + M/(1 - l'_2) - M/(1 - l'_1 l'_2)$, and from Lemma 2, the total number of transmissions with XORs coding is $N_b = M/\min\{1 - l'_1, 1 - l'_2\}$, where $l'_i = 1 - (1 - l_i)(1 - s_i)$.

Without loss of generality, suppose that $l'_1 \geq l'_2$ and $l'_2 = \beta l'_1$, $0 \leq \beta \leq 1$. We have

$$\begin{aligned}
 \gamma &= \frac{N_a - N_b}{N_a} \\
 &= 1 - \frac{N_b}{N_a} \\
 &= 1 - \frac{1/(1 - l'_1)}{1/(1 - l'_1) + 1/(1 - l'_2) - 1/(1 - l'_1 l'_2)} \\
 &= 1 - \frac{1}{1 + (1 - l'_1)/(1 - \beta l'_1) - (1 - l'_1)/(1 - \beta l'^2_1)},
 \end{aligned} \tag{C.3}$$

Define a function $f(\beta) = \gamma$ with β being the variable. We can easily prove that $f(\beta)$ is an increasing function. Thus, when β is 1, the value of function $f(\beta)$ is maximum. That is when $l'_1 = l'_2$, the network coding gain γ is maximum, which proves our Corollary 1.

D. Proof of Lemma 3

From the analysis in the previous section, we can see that the total number of active time slots consumed for data dissemination with XORs coding is

$$\begin{aligned}
 \eta_s &= N_b \sum_{i=1}^n (1 - s_i) \\
 &= \frac{M \sum_{i=1}^n (1 - s_i)}{\min_{i \in \{1, 2, \dots, n\}} \{(1 - l_i)(1 - s_i)\}},
 \end{aligned} \tag{D.4}$$

where s_i is the probability that sensor p_i is in sleep mode at each time slot.

However, if there is no sleep scheduling at sensors, that is $s_i = 0$, the total number of transmissions for disseminating sufficient large M packets to n receivers with XORs coding is

$$N'_b = \frac{M}{1 - \max_{i \in \{1, 2, \dots, n\}} \{l_i\}}. \tag{D.5}$$

Since no sensors are in sleep mode, the total number of active time slots consumed for disseminating packets with XORs coding is

$$\begin{aligned}
 \eta_{ns} &= N'_b n \\
 &= \frac{Mn}{1 - \max_{i \in \{1, 2, \dots, n\}} \{l_i\}}.
 \end{aligned} \tag{D.6}$$

From the above formulation, we know that only if $\eta_s < \eta_{ns}$, sleep scheduling has contribution to save energy consumed by *idle listening*, otherwise, the retransmission due to sleep scheduling in sensors imposes more energy consumption. The above $\eta_s < \eta_{ns}$ changes into

$$\frac{M \sum_{i=1}^n (1 - s_i)}{\min_{i \in \{1, 2, \dots, n\}} \{(1 - l_i)(1 - s_i)\}} < \frac{Mn}{1 - \max_{i \in \{1, 2, \dots, n\}} \{l_i\}}. \tag{D.7}$$

That is

$$\sum_{i=1}^n (1 - s_i) < \frac{n}{l_{\min}} \min_{i \in \{1, 2, \dots, n\}} \{(1 - l_i)(1 - s_i)\}, \tag{D.8}$$

where $l_{\min} = 1 - \max_{i \in \{1, 2, \dots, n\}} \{l_i\}$.

Thus, if (D.8) can be satisfied, the current sleep scheduling must have contribution to save energy compared with no sleep scheduling.

Acknowledgment

This paper is supported by the National Science Foundation of China under Grant no. 60773036.

References

- [1] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 81–94, Baltimore, Md, USA, November 2004.
- [2] C.-J. M. Liang, R. Musáloiu-E., and A. Terzis, "Typhoon: a reliable data dissemination protocol for wireless sensor networks," in *Proceedings of the 5th European Conference on Wireless Sensor Networks (EWSN '08)*, vol. 4913 of *Lecture Notes in Computer Science*, pp. 268–285, January 2008.
- [3] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy, "PSFQ: a reliable transport protocol for wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 1–11, Atlanta, Ga, USA, September 2002.
- [4] S. S. Kulkarni and L. Wang, "MNP: multihop network reprogramming service for sensor networks," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS '05)*, pp. 7–16, June 2005.
- [5] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *Proceedings of the 24th IEEE Annual Joint Conference of the Computer and Communications Societies (INFOCOM '05)*, vol. 4, pp. 2470–2481, Miami, Fla, USA, 2005.
- [6] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st IEEE Annual Joint Conference of the Computer and Communications Societies (INFOCOM '02)*, vol. 3, pp. 1567–1576, 2002.
- [7] A. Cerpa, J. L. Wong, L. Kuang, M. Potkonjak, and D. Estrin, "Statistical model of lossy links in wireless sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 81–88, Los Angeles, Calif, USA, 2005.
- [8] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 914–925, 2009.
- [9] S. Y. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson, "On the minimum number of transmissions in single-hop wireless coding networks," in *Proceedings of the IEEE Information Theory Workshop (ITW '07)*, pp. 120–125, Tahoe City, Calif, USA, September 2007.
- [10] C. Zhan, Y. Xu, J. Wang, and V. Lee, "Reliable multicast in wireless networks using network coding," in *Proceedings of the 6th International Conference on Mobile Adhoc and Sensor Systems (MASS '09)*, pp. 506–515, Macau, China, 2009.
- [11] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

- [12] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 829–848, 2006.
- [13] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *Computer Communication Review*, vol. 36, no. 1, pp. 63–68.
- [14] S. Deb, M. Effros, and T. Ho, "Network coding for wireless applications: a brief tutorial," in *Proceedings of the International Workshop on Wireless Adhoc Networks (IWVAN '05)*, May 2005.
- [15] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability gain of network coding in lossy wireless networks," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 2171–2179, Phoenix, Ariz, USA, April 2008.
- [16] Y. Lin, B. Liang, and B. Li, "Data persistence in large-scale sensor networks with decentralized fountain codes," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1658–1666, Anchorage, Alaska, USA, May 2007.
- [17] I.-H. Hou, Y.-E. Tsai, T. F. Abdelzaher, and I. Gupta, "Adap-Code: adaptive network coding for code updates in wireless sensor networks," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 1517–1525, Phoenix, Ariz, USA, 2008.
- [18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.
- [19] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sen Gupta, "Loss-aware network coding for unicast wireless sessions: design, implementation, and performance evaluation," in *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '08)*, pp. 85–96, Annapolis, Md, USA, 2008.
- [20] F.-C. Kuo, K. Tan, X. Y. Li, J. Zhang, and X. Fu, "XOR rescue: exploiting network coding in lossy wireless networks," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '09)*, pp. 1–9, Rome, Italy, June 2009.
- [21] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in wireless networks," in *Proceedings of the 20th IEEE Annual Joint Conference of the Computer and Communications Societies (INFOCOM '01)*, pp. 1028–1037, 2001.
- [22] K. Iwanicki and M. Van Steen, "Multi-hop cluster hierarchy maintenance in wireless sensor networks: a case for gossip-based protocols," in *Proceedings of the 6th European Conference on Wireless Sensor Networks*, pp. 102–117, Springer, Cork, Ireland, February 2009.
- [23] G. Hartl and B. Li, "Loss inference in wireless sensor networks based on data aggregation," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 396–404, April 2004.
- [24] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '07)*, pp. 169–180, Kyoto, Japan, 2007.