Lifetime-Resource Tradeoff for Multicast Traffic in Wireless Sensor Networks

Vahid Shah-Mansouri, Student Member, IEEE and Vincent W.S. Wong, Senior Member, IEEE

Abstract—In this paper, we study the problem of supporting multicast traffic in wireless sensor networks with network coding. On one hand, coding operations can reduce power consumption and consequently improve the network lifetime. On the other hand, performing network coding requires the use of the limited resources of the sensor nodes such as memory and energy. We study the tradeoff between maximizing the network lifetime and minimizing the number of network coding operations. We introduce the coding flow variables which enable us to determine the rate at which different operations (e.g., forwarding, replication, coding) are performed in each sensor node. Using the coding flow variables, we formulate the maximum-lifetime minimum-resource (MLMR) coding subgraph problem as a linear programming problem. The objective in MLMR problem is to jointly maximize the network lifetime and minimize the rate of performing network coding. We propose an MLMR algorithm in order to obtain the optimal coding subgraph. We investigate the lifetime-resource tradeoff assuming that the cost of performing network coding varies for intermediate nodes. Simulation results show that the network lifetime can considerably be improved when the cost of performing network coding is relatively low compared to the case that this cost is high for intermediate nodes in the network. Moreover, results show that the network lifetime can substantially be increased using MLMR algorithm compared with the classical multicast with Steiner tree and another algorithm which uses network coding without considering the broadcast nature of wireless links.

Keywords: multicast traffic, wireless sensor networks, network coding, lifetime maximization.

I. INTRODUCTION

Network coding refers to the notion of performing coding (e.g., binary addition, XOR) on the content of the packets at intermediate network devices. A network device performs network coding when it combines incoming packets which have to reach different sinks and creates a coded packet which has to reach the same set of sinks. For multicast and broadcast traffic, network coding can lead to substantial improvement in network throughput of wired and wireless networks [1]–[6]. It has been shown in [1] that the max-flow min-cut bound on multicast capacity is achievable if nodes are allowed to perform network coding. The problem of minimum cost multicast for both wired and wireless networks is studied in [7]. The wireless network is modeled as a hypergraph, which represents the broadcast nature of the wireless links. The performance of network coding using hypergraph model is investigated in [8]. In addition to the throughput gain, network coding can reduce power consumption and congestion by reducing the total number of packets transmitted in the network for unicast and multicast traffic. Different upper bounds have been derived in [9]–[12] for the energy gain of network coding.

The problem of supporting multicast over coded packet networks can be divided into two disjoint design problems. The first problem determines the nodes in the network that perform network coding, which is also referred to as the problem of determining the coding subgraph in the literature. The second problem determines the code that should be used on the coding subgraph. A linear programming formulation is used in [7] to construct the minimum cost coding subgraph. A distributed algorithm is proposed in [13] to construct the maximum utility coding subgraph. It has been shown that random codes are useful in achieving the desired capacity and facilitating distributed implementation [14], [15].

For the problem of determining the minimum resource coding subgraph, results in [16] show that for acyclic networks with two sources, the number of nodes required to perform network coding is always less than the number of sinks. However, the results cannot be extended to an arbitrary number of sources. The problem of maximizing network throughput while using the minimum set of nodes to perform network coding is inherently an NP-hard problem. In [17], a heuristic based on genetic algorithm is used to find the sub-optimal solution. In [18], the problem of minimizing network coding resources for wired networks is formulated as a linear programming problem. The objective is to minimize the rate at which packets undergo coding in the network. Although this method can provide a distributed solution, it cannot be applied to wireless networks with broadcast links.

The problem of constructing the multicast tree for routed packet networks is equivalent to the problem of finding the Steiner tree, which is NP-complete. Due to the complexity of this problem, several heuristics are proposed. A multicast incremental power algorithm is proposed in [19] to construct a multicast tree for wireless networks while exploiting the broadcast nature of the wireless links. Interestingly, the problem of constructing the coding subgraph can be formulated as a linear programming problem for coded packet networks. This problem can be solved in polynomial time and it is suitable for distributed implementation.

The focus of our paper is on constructing the coding subgraph for multicast traffic in wireless sensor networks (WSNs). WSNs consist of battery-powered low cost wireless sensor nodes. Sensor nodes typically have limited energy
supply, memory and computational capability. Various energy-efficient medium access control and routing protocols have been proposed for WSNs (e.g., [20], [21]). Sensor devices performing network coding require additional capabilities. For example, they need to have more computational capability and memory to perform coding, especially when there is a large number of sessions in the network. Moreover, further complexity is incurred and more power is consumed when network coding is performed in the application layer. Current sensor devices either do not have these capabilities or they can be costly to implement. Hence, it is advantageous to either minimize the rate of network coding operation or limit the number of devices performing network coding in the network.

The problem of choosing the minimum set of nodes to perform network coding is NP-hard [17] in general. However, we show that the problem of minimizing the rate of network coding operations can be formulated as a linear programming problem.

In this paper, we formulate the maximum-lifetime minimum-resource (MLMR) coding subgraph problem to support multicast traffic in WSNs with network coding. We propose an MLMR algorithm to determine the rate of performing network coding in each sensor node. The contributions of this paper are as follows:

- We propose the coding flow variables, which are new representation for the set of information flow and actual rate variables. The coding flow variables enable us to determine the type of operation(s) (i.e., forwarding, replication, or network coding) performed in each wireless sensor node. These variables can also determine the rate of performing network coding and replication in each sensor node.

- We formulate a linear programming optimization problem for joint maximization of the network lifetime and minimization of the rate of performing network coding. We propose the MLMR algorithm which enables us to study the tradeoff between maximizing the network lifetime and minimizing the rate of coding operations.

- We present two model extensions including the case when the objective is to minimize the number of sensors performing network coding and the case when only a fixed number of sensor nodes are capable of performing network coding.

- We investigate the lifetime-resource tradeoff in the network. Simulation results show that the network lifetime can considerably be improved when the cost of performing network coding is relatively low compared to the case that this cost is high for intermediate nodes in the network.

The rest of this paper is organized as follows: In Section II, we introduce the notations and system model. In Section III, we describe our proposed coding flow variables and formulate the joint maximum lifetime minimum resource coding subgraph problem. We present our proposed MLMR algorithm. The model extensions are also described. In Section IV, we evaluate the performance of our proposed MLMR algorithm and compare it with three other algorithms [18], [22], [23]. Conclusions are given in Section V.

II. SYSTEM MODEL

We represent the WSN as a hypergraph, which can effectively model the broadcast nature of wireless links. Let \( H = (V, A) \) denote a directed hypergraph, where \( V \) is the set of nodes and \( A \) is the set of hyperarcs. Let \( N_i \) denote the set of neighbors of node \( i \in V \). A hyperarc is a pair \((i, J)\) representing a broadcast link from node \( i \) to the subset of neighboring nodes \( J \subset N_i \). Let \( \mathcal{I}_i \) denote the power set (i.e., the set of all subsets of a given set) of set \( N_i \) except the empty set. The set of hyperarcs in the network can be represented as

\[
A = \{(i, J) \mid i \in V, J \in \mathcal{I}_i\}.
\]

Let \( \mathcal{A}^{in}_i \) denote the set of all incoming hyperarcs of node \( i \in V \). These are the hyperarcs including node \( i \) as the destination:

\[
\mathcal{A}^{in}_i = \{(j, \mathcal{I}) \mid j \in V, \mathcal{I} \in \mathcal{I}_j, i \in \mathcal{I}\}.
\]

The cardinality of hyperarc \((i, J)\) is equal to \(|J|\), which is the number of elements in set \( J \). Let \( S \) denote the set of source nodes in the network and \( D^s \) denote the set of sinks to which source \( s \in S \) sends data. We call source \( s \) with its respective sinks as session \( s \). Let \( g^{sd}_{ij} \) denote the information flow rate of session \( s \) from node \( i \) to node \( j \in J \) through hyperarc \((i, J)\) towards sink \( d \in D^s \). Let \( z^s_{ij} \) denote the actual rate at which the data of session \( s \) is sent on hyperarc \((i, J)\). This variable represents the data rate sent on hyperarc \((i, J)\) after performing coding at node \( i \). The actual rate vector \( z = \{z^s_{ij}\} \) is called the coding subgraph of the network.

When node \( i \) performs network coding on packets of session \( s \), it combines several incoming packets of this session destined for different sinks and transmits one coded packet. Such a coded packet should reach all the sinks that the original packets had been destined for. This happens by means of replication on the route of the coded packet to the sinks of session \( s \). Node \( i \) sends a packet on hyperarc \((i, J)\) with cardinality greater than one if the packet needs to reach different sinks through different neighbors in set \( J \). In case that the coded packet is sent on a hyperarc and this coded packet traverses different routes to the sinks, replication is not performed. The following lemma shows that it is not necessary to consider all the hyperarcs in the network graph to obtain the coding subgraph.

**Lemma 1.** It is not necessary to consider hyperarcs with cardinality greater than \(|D^s|\) (i.e., the number of sinks of session \( s \)) to obtain the coding subgraph of session \( s \).

Proof. For session \( s \in S \), let \( b^{D^s}_{iJ} \) denote a packet sent on the hyperarc \((i, J)\) destined for the sinks in \( D \in D^s \). For node \( i \) and outgoing hyperarc \( J \in \mathcal{I}_i \), the coded packet \( b^{D^s}_{iJ} \) is created by combining several packets as follows:

\[
\begin{align*}
  b^{D^s}_{iJ} &= \sum_{j \in J} \omega_{ij} b^{sD^s}_{ij} \quad (1)
\end{align*}
\]

where \( b^{sD^s}_{ij} \) is a packet of session \( s \) destined for the set of destinations in \( D^s \) and it is sent from node \( i \) to \( j \) if hyperarc \((i, J)\) is not used. If \(|D^s| > 1\), then the packet \( b^{sD^s}_{ij} \) is a coded packet. \( \omega_{ij} \) is the coding coefficient (chosen from a
Galois field) that node \( i \) used for packet \( k_{ij}^D \). We notice that set \( D \) is the union of sets \( D_j \) (i.e., \( D = \bigcup_{j \in J} D_j \)). Since the packets destined for the same destinations cannot be combined, the sets \( D_j \) should be disjoint for different \( j \in J \). The number of such packets is \( |J| \) while the number of destinations in session \( s \) is \( |D^s| \). Hence, the maximum number of packets which are combined can be at most \( |D^s| \) (i.e., \(|J| \leq |D^s|\)). Therefore, node \( i \) does not send any packet on hyperarc \((i, J)\) if \(|J| > |D^s| \) (i.e., \( z_{ij}^D = 0 \)). Consequently, it is not necessary to consider these hyperarcs to obtain the coding subgraph of session \( s \).

Based on Lemma 1 and the definition of \( A, A_{in}^s \), and \( I_i \), for session \( s \), we define the following sets:

\[
\tilde{A}^s = \{(i, J) \mid (i, J) \in A, \ |J| \leq |D^s|\},
\]

\[
\tilde{A}_{in}^s = \{(j, I) \mid (j, I) \in A_{in}^s, \ |I| \leq |D^s|\}, \quad i \in V\]

\[
\tilde{I}_i^s = \{J \in I_i \mid |J| \leq |D^s|\}, \quad i \in V.
\]

In WSNs, each node has a large number of neighboring nodes while the number of sinks gathering information in each session is limited. Therefore, the number of variables in the system can be considerably reduced by removing the hyperarcs with cardinality greater than \(|D^s|\).

**A. System Constraints**

There are several constraints in our problem formulation. Let \( r_s \) denote the data generation rate by source node \( s \in S \). The flow conservation constraint in node \( i \in V \) for data sent from source \( s \) towards sink \( d \in D^s \) is

\[
\sum_{J \in I_i^s} \sum_{j \in J} g_{ij}\hat{d}_{ij} - \sum_{(j, I) \in \tilde{A}_{in}^s} g_{jI} = \beta_i,
\]

where

\[
\beta_i = \begin{cases} r_s, & \text{if } i = s, \\ -r_s, & \text{if } i = d, \\ 0, & \text{otherwise}. \end{cases}
\]

This constraint guarantees that the outgoing information flow rate from node \( i \) is equal to the input rate plus the rate of information generated by node \( i \) for session \( s \).

By the flow-sharing property of network coding and the broadcast nature of wireless channel, the actual flow rate on each hyperarc needs only be the maximum of the individual sinks flows. The actual rate constraint for session \( s \) on hyperarc \((i, J)\) can be expressed as [7]

\[
z_{ij}^D \geq \sum_{j \in J} g_{ij}\hat{d}_{ij}, \quad \forall d \in D^s.
\]

We extend the protocol interference model in multi-hop wireless networks for hypergraph. Let \( \delta(i, j) \) and \( \kappa(i, j) \) denote the communication range and interference range between nodes \( i \) and \( j \), respectively. Let \( \delta(i, J) = \max_{j \in J} \delta(i, j) \) and \( \kappa(i, J) = \max_{j \in J} \kappa(i, j) \) denote the communication range and interference range between node \( i \) and the farthest node in set \( J \), respectively. Two hyperarcs \((i_1, J_1)\) and \((i_2, J_2)\) interfere with each other (i.e., cannot be active simultaneously) if any of the following conditions are satisfied: (a) \( i_1 = i_2 \), (b) there exists \( j \in J_1 \) such that \( \delta(i_2, j) \leq \kappa(i_1, J_1) \), (c) there exists \( j \in J_2 \) such that \( \delta(i_1, j) \leq \kappa(i_2, J_2) \).

The concept of contention graph and maximal clique [24] can also be extended for a hypergraph. In a contention hypergraph, each node corresponds to a hyperarc. There is an edge between two hyperarcs if they cannot be active simultaneously. The maximal cliques in a hypergraph can be determined accordingly. Let \( C \) denote the ordered set of all maximal cliques, with \( C_k \) representing the \( k \)th element in set \( C \). The link scheduling constraints are

\[
\sum_{(i, J) \in C_k} x_{ij}^D \leq \lambda_k, \quad k = 1, \ldots, |C|
\]

where \( \lambda_k \) represents the fraction of time that hyperarcs in set \( C_k \) can be active, \( c_{ij} = \min_{j \in J} c_{ij} \) is the capacity of hyperarc \((i, J)\), and \( c_{ij} \) is the capacity of the link between node \( i \) and \( j \).

The network lifetime is defined as the time when the first node runs out of its energy [20, 25, 26]. Whenever a sensor node is no longer operational, the coding subgraph is updated based on the new network topology and the remaining energy of the active sensor nodes. Nodes employ power control to save energy. The transmission power of a sender is adjusted based on the distance of the two nodes. Let \( E_i \) denote the initial energy of node \( i \) and \( p_{ij} \) represent the power consumed to transmit one bit of data on hyperarc \((i, J)\). This is equal to the power consumed to transmit data to the farthest node in set \( J \). Thus, \( p_{ij} = \min_{j \in J} p_{ij} \). Let \( p_{ij} \) be the power consumed to transmit one bit of data from node \( i \) to \( j \). For node \( j \) to receive the bit successfully, it requires that the received power \( p_r = p_{ij}/d(i, j)^\alpha > \eta_p \), where \( d(i, j) \) is the distance between nodes \( i \) and \( j \), \( \alpha \) is the path loss exponent with \( 2 \leq \alpha \leq 4 \), and \( \eta_p \) is a threshold value. The total transmit power is given by [20, 22, 25, 26]:

\[
p_{ij} = \eta_1 + \eta_2 d(i, j)\eta_3,
\]

where \( \eta_1 \) is the power consumed in the transmitter circuitry. Given the maximum transmission power \( p_{max} \), the corresponding communication range \( \delta(i, j) \) can be determined.

Given \( E_i \) and \( p_{ij} \), the lifetime of node \( i \) under actual rate vector \( z = \{z_{ij}^D\} \) is

\[
\Gamma_i(z) = \frac{E_i}{\sum_{s \in S} \sum_{J \in \tilde{I}_i^s} p_{iJ} z_{ij}^D}, \quad \forall i \in V.
\]

Let \( q \) denote an upper bound on the inverse of the lifetime of all the nodes (i.e., \( 1/\Gamma_i(z) \leq q \), for all \( i \in V \)). Minimizing the variable \( q \) is equivalent to maximizing the minimum node’s lifetime in the network (i.e., \( \max \min_{i \in V} \Gamma_i(z) = \min q \)). Using equation (9), the network lifetime constraint can be expressed as

\[
\sum_{s \in S} \sum_{J \in \tilde{I}_i^s} p_{iJ} z_{ij}^D \leq E_i q, \quad \forall i \in V.
\]

Considering the constraints introduced on the information flow rate and the actual rate variables, the coding subgraph
of the network is characterized in [7, Theorem 2], which we reproduce here, slightly adapted:

Theorem 1. The actual rate vector \( z = \{ z^s_{ij} \} \) is part of the set constructed by the constraints (5)–(8) and (10) if and only if there exists a network code for each session \( s \) that sets up a multicast connection at a rate arbitrarily close to \( r_s \).

Since designing codes for optimal inter-session coding for practical wireless networks is an active and open research area, we restrict network coding operation within sessions (i.e., intra-session coding) in this paper. The only relation between various sessions is imposed by the link scheduling constraint in (7)–(8). In practice, random linear network coding can achieve the rate of multicast connection \( r_s \) while it can be performed distributively in the network. Nodes on the coding subgraph which are responsible for network coding select the coding coefficients randomly from a Galois field \( \mathbb{GF}(2^r) \) and combine the incoming packets with the corresponding rate. Sinks can decode the received packets with probability close to one for large values of \( u \) [27].

III. PROBLEM FORMULATION

Our goal is to formulate the joint problem of maximizing the network lifetime and minimizing the rate of performing coding operations. On one hand, Higher rates of coding operations require more resources and capabilities in the system. On the other hand, a larger number of packets undergoing network coding reduce the total number of packet transmissions in the network. This decreases the average power consumption. Consequently, network lifetime can be increased when the rate of coding operations increases. To formulate this problem, it is necessary to determine the rate of coding operations at each network device. The information flow and actual rate variables alone are not sufficient to determine the corresponding rates on the incoming and outgoing hyperarcs of each node. In this section, we introduce a new set of variables which allows us to determine the type of the operations and the rate at which these operations are performed by each node. Let \( \mathcal{P}^s \) denote the power set of \( \mathcal{D}^s \) except the empty set. For simplicity, we label the sinks of session \( s \) by positive integers. For example, if session \( s \) has three sinks, then \( \mathcal{D}^s = \{1, 2, 3\} \) and \( \mathcal{P}^s = \{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\} \). For hyperarc \((i, \mathcal{J})\), we fix the ordering of the nodes in \( \mathcal{J} \). Let \( j_m \) denote the \( m \)th member of \( \mathcal{J} \).

We now introduce the coding flow variables \( n^s_{ij} \). We show that using this variable, one can determine the rate of performing coding at each sensor node. For session \( s \), hyperarc \((i, \mathcal{J})\), and \( P_1, \ldots, P_{|\mathcal{J}|} \in \mathcal{P}^s \), let \( n^s_{ij}(P_1, \ldots, P_{|\mathcal{J}|}) \) represent the rate of transmission of packets which are forwarded towards sinks in set \( P_m \) by neighbor \( j_m \) for \( m = 1, \ldots, |\mathcal{J}| \). These packets have to reach the sinks in set \( \bigcup_{m=1}^{\mathcal{J}} P_m \) via different neighbors of node \( i \). The size of the coding variable for each hyperarc depends on the cardinality of the hyperarc. For example, for hyperarc \((i, \mathcal{J})\) with cardinality two, the coding flow variable is \( n^s_{ij}(P_1, P_2) \), where \( P_1 \) and \( P_2 \in \mathcal{P}^s \).

For illustration purpose, consider Fig. 1. Node \( s \) is the source and nodes \( d_1 \) and \( d_2 \) are the sinks. Assume that node \( h_3 \) sends one packet per second towards \( d_1 \) and one packet per second towards \( d_2 \). Node \( h_3 \) can combine these packets and send one packet over hyperarc \((h_3, \mathcal{J})\) where \( \mathcal{J} = \{d_1, d_2\} \). Then, we have \( n^s_{ij}(P_1, P_2) = 1 \), where \( P_1 = \{d_1\} \) and \( P_2 = \{d_2\} \).

A coded packet is routed towards a sink only from one path. It implies that for coding flow variable \( n^s_{ij}(P_1, \ldots, P_{|\mathcal{J}|}) \), the sets \( P_1, \ldots, P_{|\mathcal{J}|} \) are disjoint (i.e., \( P_a \cap P_b = \emptyset \) for \( a, b = 1, \ldots, |\mathcal{J}|, a \neq b \)). Let \( P^s_{ij} \) denote the set of all disjoint subsets of \( \mathcal{P}^s \) with \(|\mathcal{J}|\) members. Let \( P^s_{ij}(d, m) \) denote the set of members of \( P^s_{ij} \) while \( d \in \mathcal{P}_m \). These can be defined as follows:

\[
P^s_{ij} = \{(P_1, \ldots, P_{|\mathcal{J}|}) | P_1, \ldots, P_{|\mathcal{J}|} \in \mathcal{P}^s \}
\]

and

\[
P^s_{ij}(d, m) = \{(P_1, \ldots, P_{|\mathcal{J}|}) | P_1, \ldots, P_{|\mathcal{J}|} \in \mathcal{P}^s \mid d \in \mathcal{P}_m \}
\]

For example, consider hyperarc \((i, \mathcal{J})\) with cardinality two (i.e., \( \mathcal{J} = \{j_1, j_2\} \)) and session \( s \) with two sinks, namely sink 1 and 2. Then, we have \( P^s_{ij} = \{(\{1\}, \{2\}), (\{2\}, \{1\})\} \) for \( P^s_{ij} = \{(\{1\}, \{2\}), (\{2\}, \{1\})\} \) and \( P^s_{ij} = \{(\{2\}, \{1\})\} \).

We now show how to relate the information flow variables \( g^s_{ij} \) to the coding flow variables \( n^s_{ij} \). For simplicity, we assume that the size of each packet is \( M \) bits. The information flow variable \( g^s_{ij} \) is equal to the packet size (i.e., \( M \)) times the total rate at which packets which have to reach sink \( d \) of session \( s \) through neighbor \( j_m \) (i.e., \( \forall n^s_{ij}(P_1, \ldots, P_{|\mathcal{J}|}) \) while \( d \in \mathcal{P}_m \)). The relation of coding flow and information flow variables can be expressed as

\[
g^s_{ij} = M \sum_{(P_1, \ldots, P_{|\mathcal{J}|}) \in P^s_{ij}(d, m)} n^s_{ij}(P_1, \ldots, P_{|\mathcal{J}|})
\]

For illustration purpose, consider the following example. Assume that there are three sinks in session \( s \), namely 1, 2, 3. For hyperarc \((i, \mathcal{J})\) with cardinality two (i.e., \( \mathcal{J} = \{j_1, j_2\} \)), equation (13) can be written for sink 1 and neighbors \( j_1 \) and \( j_2 \) as follows:

\[
g^s_{ij} = \sum_{(P_1, P_2) \in P^s_{ij}(1, 1)} n^s_{ij}(P_1, P_2)
\]

\[
g^s_{ij} = \sum_{(P_1, P_2) \in \mathcal{P}^s \mid P_1 \cap P_2 = \emptyset, i \in P_1} n^s_{ij}(P_1, P_2)
\]

\[
= n^s_{ij}(\{1\}, \{2\}) + n^s_{ij}(\{1\}, \{3\}) + n^s_{ij}(\{1\}, \{2, 3\}) + n^s_{ij}(\{2\}, \{3\}) + n^s_{ij}(\{1, 2\}, \{3\}) + n^s_{ij}(\{1, 2, 3\})
\]

Fig. 1. Sample network with the source node \( s \) and two sinks \((d_1, d_2)\).
The coding flow variable over hyperarc variables \( i \) \( J \) is the summation of the rate of packet transmissions towards all the sinks (i.e., all the coding flow variables of that hyperarc multiplied by the packet size) as:

\[
z_i^{s, J} = \frac{g_i^s J}{M} = \sum_{(P_1, P_2) \in P_i^{s, J}(1, 2)} n_i^{s, J}(P_1, P_2) = \sum_{(P_1, P_2) \in P_i^{s, J}} n_i^{s, J}(P_1, P_2) = n_i^{s, J}([2], [1]) + n_i^{s, J}([2], [1]) + n_i^{s, J}([2], [1]) + n_i^{s, J}([2], [1]) + n_i^{s, J}([2], [1]) + n_i^{s, J}([2], [1]) + n_i^{s, J}([2], [1]).
\]

The actual rate of hyperarc \( i, J \) is the summation of the rate of packet transmissions towards all the sinks (i.e., all the coding flow variables of that hyperarc multiplied by the packet size) as:

\[
z_i^{s, J} = M \sum_{(P_1, \ldots, P_\mid J \rangle) \in P_i^{s, J}} n_i^{s, J}(P_1, \ldots, P_\mid J \rangle).
\]

Consider the following example. Assume that there are three sinks in session \( s \), namely \( 1, 2, 3 \). Consider hyperarc \((i, J)\) with cardinality two (i.e., \( J = \{j_1, j_2\} \)). The actual rate on this hyperarc is the summation of all coding flow variables multiplied by the packet size as follows:

\[
z_i^{s, J} = \sum_{j \in J} n_i^{s, J}(\{1\}, \{2\}) + n_i^{s, J}(\{1\}, \{3\}) + n_i^{s, J}(\{2\}, \{1\}) + n_i^{s, J}(\{2\}, \{3\}) + n_i^{s, J}(\{3\}, \{1\}) + n_i^{s, J}(\{3\}, \{2\}) + n_i^{s, J}(\{2\}, \{1\}).
\]

Fig. 2 shows a sample network with three sinks and one source node. The source is generating information with a rate of 2 units per second, and the data is sent towards all three sinks. The packet size is assumed to be one. The information flow, actual rate, and coding flow values of all hyperarcs plus the paths that packets travel are shown on each hyperarc.

Equations (13) and (16) imply that the information flow and actual rate variables are dependent variables while the coding flow variables are independent variables. We keep these dependent variables since they make the presentation of the formulation clear and easier to understand.

The coding flow variables provide information on the packet transmission rates for different disjoint sets of sinks on each hyperarc. We can determine the type of operation performed in node \( i \) by inspecting variables \( n_i^{s, J} \). Let \( \phi_i^s(R) \) denote the total rate at which node \( i \) sends coded packets which have to reach sinks in set \( R \in P^s \) on the outgoing hyperarcs. Let \( \phi_i^s(J) \) denote the rate at which these packets are sent on hyperarc \((i, J)\). Variable \( \phi_i^s(R) \) is the summation of variables \( \phi_i^s(J) \) over all outgoing hyperarcs of node \( i \) (i.e., \((i, J) \in \tilde{I}^s_i \)). The coding flow variable \( n_i^{s, J}(P_1, \ldots, P_\mid J \rangle) \) is defined as the rate at which packets which have to reach sinks in set \( \cup_m P_m \) are sent over hyperarc \((i, J)\). Therefore, the rate at which node \( i \) sends coded packets which have to reach sinks in set \( R \in P^s \) over hyperarc \((i, J)\) is equal to the summation of coding flow variables \( n_i^{s, J}(P_1, \ldots, P_\mid J \rangle) \) such that \( \cup_m P_m = R \). This can be expressed as

\[
\phi_i^s(R) = \sum_{(i, J) \in \tilde{I}^s_i} \phi_i^s(J) = \sum_{(i, J) \in \tilde{I}^s_i} \sum_{(P_1, \ldots, P_\mid J \rangle) \in P_i^{s, J}} n_i^{s, J}(P_1, \ldots, P_\mid J \rangle).
\]

Let \( \psi_i^s(R) \) denote the rate at which node \( i \) receives coded packets which have to reach the sinks in set \( R \in P^s \). This is equal to the summation of rate of those packets over all incoming hyperarcs of node \( i \) (i.e., \((j, I) \in \tilde{A}^s_i \)). For hyperarc \((j, I)\), which is an incoming hyperarc of node \( i \), the coding variable \( n_j^s(I) \) contains the information of sinks in set \( R \) for node \( i \) if there exists \( v \in \{1, \ldots, |I|\} \) such that \( i = v \), i.e., \( i \) is the \( v \) th member of set \( I \) and \( P_v = R \). Therefore, we have

\[
\psi_i^s(R) = \sum_{(j, I) \in \tilde{A}^s_i \cap \tilde{A}^s_i} n_j^s(I) \sum_{v \in \{1, \ldots, |I|\} \mid P_v = R, i = v} n_j^s(P_1, \ldots, P_\mid I \rangle).
\]

Variables \( \phi_i^s(R) \) and \( \psi_i^s(R) \) represent the rate at which node \( i \) sends and receives packets destined for the sinks in set \( R \). The value of \( \phi_i^s(R) \) is less than \( \psi_i^s(R) \) if node \( i \) combines received packets destined for sinks in set \( R \) and creates a coded packet. On the other hand, the value of \( \phi_i^s(R) \) is greater than \( \psi_i^s(R) \) if node \( i \) replicates a coded packet towards the sinks in set \( R \). From these variables, we can determine the rate at which node \( i \) performs coding or replication on the packets destined for different sets of sinks. Let \( T^s \) denote the set of all collections of two or more disjoint subsets of \( P^s \). Any replication or coding operation performed at each node corresponds to one of the members of \( T^s \). Node \( i \) performs replication with respect to set \( T \in T^s \) if it transmits \( |T| \) copies of a coded packet to \( |T| \) different neighbors. Each copy is transmitted towards the sinks in a member of \( T \). Node \( i \) performs coding with
respect to set $T$ if it combines $|T|$ incoming packets which have to reach sinks in members of $T$ and sends one coded packet. For example, for a network with three sinks, we have $T^s = \{\{1\}, \{2\}, \{1, 2\}\}, \{\{1\}, \{3\}\}, \{\{2\}, \{3\}\}, \{\{1\}, \{2, 3\}\}, \{\{2\}, \{1, 3\}\}, \{\{3\}, \{1, 2\}\}, \{\{1\}, \{2, 3\}\}$. If node $i$ receives three packets which have to reach sinks 1, 2, and 3 and creates one coded packet which have to reach the sinks in set $\{1, 2, 3\}$, then it performs coding with respect to set $\{\{1\}, \{2\}, \{3\}\}$. As another example, assume that node $i$ receives one packet which has to reach the sinks in set $\{1, 2, 3\}$ and replicates two packets. One of the packets is sent towards sinks $\{1, 2\}$ and the other one is sent towards sink 3. In this case, node $i$ performs replication with respect to set $\{\{1, 2\}, \{3\}\}$.

Let $\mu_i^s(T)$ and $\nu_i^s(T)$ denote the rate of performing replication and network coding with respect to set $T \in T^s$, respectively. The final step in determining the rate of performing network coding is to relate variables $\mu_i^s(T)$ and $\nu_i^s(T)$ to variables $\phi_i^s(R)$ and $\psi_i^s(R)$. In general, for node $i \in V$, session $s \in S$, and set $R \in P^s$, we have

$$\phi_i^s(R) - \psi_i^s(R) = \sum_{\{T \in T^s \mid R \in T\}} (\mu_i^s(T) - \nu_i^s(T)) - \sum_{\{T \in T^s \mid \cup_{t \in T} = R\}} (\mu_i^s(T) - \nu_i^s(T)). \quad (19)$$

Equation (19) which quantifies the amount of coding and replication operations is a flow conservation constraint. In (19), the output rate minus the input rate is equal to the rate at which packets are generated for packets which have similar destinations. In (19), the left hand side is the difference between the output and input rate for the packets which have to reach the sinks in set $R$. On the right hand side, the term $\sum_{\{T \in T^s \mid R \in T\}} \mu_i^s(T)$ denotes the rate of replication operations creating outgoing packets that have to reach the sinks in set $R$. On the right hand side, the term $\sum_{\{T \in T^s \mid \cup_{t \in T} = R\}} \nu_i^s(T)$ denotes the rate of coding operations performed on incoming packets that have to reach the sinks in set $R$. Therefore, given $\phi_i^s(R)$ and $\psi_i^s(R)$, the values of $\mu_i^s(T)$ and $\nu_i^s(T)$ cannot be uniquely determined using this set of linear equations. We select the objective function of the joint problem such that we can find the optimal rate of coding operations among feasible solutions. The total rate of performing network coding is

$$\sum_{s \in S} \sum_{i \in V \setminus \{s\}} \sum_{T \in T^s} \nu_i^s(T).$$

Note that source $s$ does not perform coding on the packets that it generates. In the maximum lifetime minimum resource coding subgraph problem, the objective is to

$$\text{minimize } q + \gamma \sum_{s \in S} \sum_{i \in V \setminus \{s\}} \sum_{T \in T^s} \nu_i^s(T), \quad (20)$$

where $\gamma$ is the cost of performing network coding. The unit of $\gamma$ depends on the unit of $\beta_i$ in (5). The value of the tuning parameter $\gamma$ determines the effect of each part in the joint objective function. The feasible set of the joint coding subgraph problem is constructed by equations (5), (7), (8), (10), (13), and (16)–(19). Problem (21) is the maximum lifetime minimum resource coding subgraph problem. This is a linear programming problem. It can be solved by various techniques such as the simplex method.
minimize \[ q + \gamma \sum_{s \in S} \sum_{i \in \mathcal{V}(s)} \sum_{T \in \mathcal{T}} \nu_i^T(T) \]
subject to
\[ \sum_{J \subseteq J_i^s} \sum_{j \in J} g_{j}^{d}(d, m) = \beta_i, \quad \forall i \in \mathcal{V}, \quad d \in \mathcal{D}^s, \quad s \in S \]
\[ \sum_{(i, J), s} \mu_i^T(T) \leq \lambda_k, \quad k = 1, \ldots, |C|, \]
\[ \sum_{k=1}^{|C|} \lambda_k \leq 1, \]
\[ \sum_{s \in S} \sum_{J \subseteq J_i^s} p_{i, J} z_{k, J} \leq E_q, \quad \forall i \in \mathcal{V} \]
\[ \frac{g_{j}^{d}(d, m)}{Q_i^T(T)} = \sum_{(i, J), s} \mu_i^T(T) + \sum_{J \subseteq J_i^s} n_{i, J}^T(P_1, \ldots, P_{|J|}), \quad \forall (i, J) \in \mathcal{A}, \quad m = 1, \ldots, |J|, \quad d \in \mathcal{D}, \quad s \in S \]
\[ z_{k, J} = M \sum_{(i, J), s} \mu_i^T(T) + \sum_{J \subseteq J_i^s} n_{i, J}^T(P_1, \ldots, P_{|J|}), \quad \forall (i, J) \in \mathcal{A}, \quad s \in S \]
\[ \phi_i^T(R) = \sum_{J \subseteq J_i^s} \sum_{(i, J), s} n_{i, J}^T(P_1, \ldots, P_{|J|}), \quad \forall i \in \mathcal{V}, \quad s \in \mathcal{S}, \quad R \in \mathcal{P} \]
\[ \psi_i^T(R) = \sum_{J \subseteq J_i^s} \sum_{(i, J), s} n_{i, J}^T(P_1, \ldots, P_{|J|}), \quad \forall i \in \mathcal{V}, \quad s \in \mathcal{S}, \quad R \in \mathcal{P} \]
\[ \phi_i^T(R) = \psi_i^T(R) + \sum_{T \in \mathcal{T}} \mu_i^T(T) - \sum_{T \in \mathcal{T}} \nu_i^T(T), \quad \forall R \in \mathcal{P}, \quad i \in \mathcal{V}, \quad s \in \mathcal{S} \]
\[ \frac{n_{i, J}^T(P_1, \ldots, P_{|J|})}{\nu_i^T(T)} \geq 0, \quad \mu_i^T(T) \geq 0, \quad \nu_i^T(T) \geq 0, \quad \lambda_k \geq 0, \quad q \geq 0, \quad \forall (i, J) \in \mathcal{A}, \quad s \in \mathcal{S}, \quad k = 1, \ldots, |C|. \]

From Theorem 1, it follows that the solution of problem (21) is the maximum lifetime minimum resource coding subgraph for an asymptotically achievable rate for multicast sessions. Although all the nodes are capable of performing network coding, only some of them perform network coding in the optimal coding subgraph.

In problem (21), only variables \( n_{i, J}^s \), \( q \), \( \mu_i^T \), \( \nu_i^T \), and \( \lambda_k \) are independent variables. Variables \( g_{j}^{d}(d, m) \), \( z_{k, J} \), \( \phi_i^T \), and \( \psi_i^T \) are auxiliary variables used to simplify the notation and formulation. Also, the second, third, fourth, and fifth constraints can be removed if the variables \( g_{j}^{d}(d, m) \), \( z_{k, J} \), \( \phi_i^T \), and \( \psi_i^T \) are replaced by variable \( n_{i, J}^s \), accordingly. Thus, the number of constraints and variables of the problem can further be reduced. The number of constraints in problem (21) and the number of independent variables \( n_{i, J}^s \), \( \mu_i^T \), \( \nu_i^T \), and \( \lambda_k \) increase linearly by the number of sessions \( |S| \) and the number of nodes \( |\mathcal{V}| \), and they increase exponentially by the number of the sinks \( |\mathcal{D}| \) of each session \( s \in \mathcal{S} \). Thus, our formulation is suitable for networks with a large number of sessions while each session has a small number of sinks. Algorithm 1 below is the MLMR algorithm for determining the coding subgraph in the network.

**Algorithm 1 - MLMR Algorithm**

1. Collect topology information (including \( \mathcal{V} \) and \( \mathcal{A} \)), set of sources \( \mathcal{S} \), and set of sinks \( \mathcal{D}^s \).
2. Determine the ordered set of all maximal cliques \( \mathcal{C} \) in the hypergraph.
3. Set variable \( \gamma \) equal to the cost of performing network coding.
4. Solve problem (21) to obtain \( n_{i, J}^s \), \( \mu_i^T \), \( \nu_i^T \), \( \lambda_k \), and \( q \).
5. Forward the values of \( n_{i, J}^s \) to each sensor node \( i \in \mathcal{V} \).
question: What is the minimum set of sensor devices that should perform network coding in order to achieve a certain level of the network lifetime? The difference between this problem and the problem presented in the previous subsection is that the number of coding devices is not fixed.

The problem of finding the coding subgraph while the objective is to jointly maximize the lifetime and minimize the number of coding devices can be formulated with modification in problem (21). Let \( \alpha_i \) be an indicator variable which denotes whether node \( i \) performs network coding or not. If \( \alpha_i = 1 \), then node \( i \) performs network coding. This boolean variable can be mapped to \( \nu_i^\alpha \) for node \( i \) as follows:

\[
\alpha_i = \begin{cases} 
1, & \text{if } \sum_{s \in S} \sum_{T \in T_s} \nu_i^\alpha(T) > 0, \\
0, & \text{otherwise}. 
\end{cases}
\]

This problem can be formulated by (a) replacing the objective function in problem (21) by minimizing \( q + \zeta \sum_{i \in V} \alpha_i \), where \( \zeta \) is a design factor, and (b) including the following constraints in problem (21):

\[
\alpha_i \geq \frac{\sum_{s \in S} \sum_{T \in T_s} \nu_i^\alpha(T)}{H_{\text{max}}}, \quad \alpha_i \in \{0, 1\},
\]

where \( H_{\text{max}} \) is an upper bound for the rate of performing network coding at all nodes. It is a constant and is used to guarantee that \( \left( \sum_{s \in S} \sum_{T \in T_s} \nu_i^\alpha(T)/H_{\text{max}} \right) \) is always less than 1. The formulated problem is a mixed integer linear programming problem with binary variables \( \alpha_i \). It can be solved by using branch-and-bound method [28].

**IV. PERFORMANCE EVALUATION**

In this section, we evaluate the performance of our proposed MLMR algorithm and compare it with the minimal resource (EMR) algorithm [23]. For the MLST algorithm, a mixed integer linear programming model is used to construct the maximum lifetime Steiner tree. We use the MOSEK [29] optimization toolbox to solve the problem. In the MATLAB simulations, we consider a square coverage area with each side equal to 70 m. There are 40 sensor nodes randomly deployed in the coverage area. One of them is the source node. There are four sinks located at the corners of the coverage area.

The random topology has been used widely in WSNs in the literature [20], [25], [26]. Each source generates 10 kbps of information (i.e., \( r_s = 10 \text{ kbps} \), \( \forall s \in S \)). The size of each packet is 256 bits. The maximum transmission range is 10 m. The path loss exponent \( a \) is set to 4 [20]. We choose \( \eta_1 = 100 \text{ pJ/sec/m} \) and \( \eta_2 = 1 \text{ nJ/sec} \) [25]. The initial energy of each sensor node is 1 kJ. Each point is averaged over 100 simulation runs.

**A. Varying the Tuning Parameter \( \gamma \)**

In the first experiment, we determine the maximum network lifetime and the total rate of performing coding by varying the tuning parameter \( \gamma \). There are four sinks deployed at the corners of the coverage area. One sensor node is randomly selected to operate as the source transmitting data towards all the sinks. A large value of \( \gamma \) (i.e., \( \gamma >> 1 \)) implies that network lifetime is more important than the rate of performing network coding. On the other hand, a small value of \( \gamma \) (i.e., \( \gamma << 1 \)) decreases the effect of network lifetime in finding the optimal coding subgraph.

Figs. 3 and 4 show the network lifetime and the total rate of performing network coding under different values of \( \gamma \), respectively. When the value of \( \gamma \) is small, the joint maximum lifetime minimum resource coding subgraph problem is reduced to the maximum lifetime coding subgraph problem. In this case, both the network lifetime and also the rate of performing coding achieve their maximum values. For small values of \( \gamma \), the MLMR algorithm outperforms the minimal resource algorithm and the MLST algorithm by 20% and 25%, respectively. The total rate of performing network coding is 30% more in our proposed MLMR algorithm when compared with the minimal resource algorithm. When the value of \( \gamma \) increases, the rate of coding operation is decreased and eventually becomes zero when \( \gamma \) is greater than 10. In this case,
the MLST algorithm performs better than the minimal resource algorithm since MLST makes use of the broadcast nature of the wireless links while the minimal resource algorithm does not. The MLMR algorithm performs better than MLST since it uses multipath routing as well as the broadcast nature of the wireless links. Multipath routing is known to outperform single path routing in terms of energy consumption and network throughput.

Fig. 5 shows the lifetime of the network versus the network coding rates. Different points of this figure relate to different values of parameter $\gamma$ and are derived from Figs. 3 and 4. When moving along this curve from left to right, the value of $\gamma$ decreases from 10 to $10^{-4}$. A further decrease of the value of $\gamma$ can neither improve the network lifetime nor increase the rate of coding operations. This figure shows that higher rates of coding operations in the network can lead to higher network lifetime. It is because network coding can reduce the total number of packets transmitted in the network and consequently can reduce the power consumption.

B. Varying the Number of Sinks $|D^s|$  

In the second experiment, we compare the performance of our MLMR algorithm with the minimal resource and MLST algorithms in the presence of different number of sinks. One sensor node is randomly selected as the source node. The number of sinks increases in each step starting from one to four. In each step, the source node transmits data towards all the available sinks. Figs. 6 and 7 show the network lifetime and the rate of performing network coding under different number of sinks, respectively. When the network only has one sink, the problem is reduced to the unicast problem. Thus, our proposed MLMR algorithm and the minimal resource algorithm provide similar performance. These algorithms perform better than the MLST algorithm because they use multipath routing rather than single path routing as in the MLST algorithm. The MLST problem in the unicast case can be solved in polynomial time.

In a unicast problem, the rate of performing network coding is zero. On the other hand, when the number of sinks increases, the amount of data flow in the network increases. The lifetime of the network decreases and the rate of performing coding operation increases accordingly.

C. Varying the Number of Sources $|S|$  

The next experiment evaluates the performance of different algorithms when the number of sources varies in the network. There are four sinks located at the four corners of the coverage area. The number of sources is increased in each step of the simulation from 1 to 5. Each source node sends data towards the two sinks which are closer to itself. Figs. 8 and 9 show the network lifetime and the rate of performing network coding under different number of sources, respectively. When the number of sources increases, there are more packet transmissions and it causes a higher energy depletion in each sensor node. Thus, the network lifetime decreases. As the number of sources increases in the network, the rate of performing network coding is increased as well. The percentage of difference between our proposed MLMR algorithm and the MLST algorithm increases from 25% for a network with one source to 35% for a network with five source nodes. Compared with the minimal resource algorithm,
our proposed MLMR algorithm has a higher network lifetime and a higher network coding rate.

D. Comparison between MLMR and EMR Algorithms

In the last experiment, we compare the MLMR algorithm with the evolutionary minimum resource (EMR) algorithm proposed in [23]. We consider the network constructed by cascading multiple copies of the topology as shown in Fig. 10 (a). The network constructed by cascading three copies is shown in Fig. 10 (b). We consider the cascaded networks with 1, 3, 7, and 15 copies. The number of sinks in these networks are 2, 4, 8, and 16, respectively. The source node is placed at the root of each topology. Table I compares the number of links performing network coding between the MLMR and EMR algorithms for different copies of topology shown in Fig. 10 (a). Since the multicast rates are achievable without network coding, the minimum number of links performing network coding is zero. Our proposed MLMT algorithm achieves the optimal solution for values of $\gamma$ greater than 1. When $\gamma$ is less than $10^{-3}$, EMR algorithm performs better than the MLMR algorithm especially for larger networks. It is because our proposed MLMR algorithm maximizes the network lifetime as the main objective and minimizes the network resources as the secondary objective when $\gamma$ is small.

V. Conclusion

In this paper, we studied the tradeoff between network lifetime and system resources utilized to perform network coding in energy-constrained wireless sensor networks. We first introduced the constraints of the system. We then proposed the coding flow variables which enable us to determine the type of operation performed in each sensor node. The rate of performing network coding and replication in each node can be determined by using our proposed variables. We formulated the joint problem of constructing maximum-lifetime minimum-resource coding subgraph as a linear programming problem. This problem is suitable for partially distributed implementation via dual decomposition. The complexity of our algorithm increases linearly by the number of nodes and number of sessions, and it increases exponentially by the number of sinks. Hence, the algorithm is suitable for networks with a large number of sessions while each session has a small number of sinks. We proposed two directions for the model extension of this problem. Simulation results showed that our proposed MLMR algorithm has a better performance when compared with the minimal resource algorithm [18], the MLST algorithm [22], and the EMR algorithm [23]. For future work, one can extend the model by considering the problem of determining the optimal location of coding devices while the objective is to maximize the network lifetime. The lifetime-resource tradeoff can also be studied for networks such that...
the multicast rate cannot be achieved without using network coding.

REFERENCES


Vahid Shah-Mansouri (S’02) received the B.Sc. and M.Sc. degrees in electrical engineering from University of Tehran, Tehran, Iran in 2003 and Sharif University of Technology, Tehran, Iran in 2005, respectively. From 2005 to 2006, he was with Farinhe-Fanavar Co., Tehran, Iran. He is currently working towards the Ph.D. degree in the Department of Electrical and Computer Engineering at the University of British Columbia (UBC), Vancouver, BC, Canada. As a graduate student, he received the UBC Four Year Fellowship and UBC Faculty of Applied Science Award. His research interests are in design and mathematical modeling of RFID systems and wireless networks.

Vincent W.S. Wong (SM’07) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently an Associate Professor. His research areas include protocol design, optimization, and resource management of communication networks, with applications to the Internet, wireless networks, RFID systems, and intelligent transportation systems. Dr. Wong is an Associate Editor of the IEEE Transactions on Vehicular Technology and an Editor of KICS/IEEE Journal of Communications and Networks. He serves as TPC member in various conferences, including IEEE Infocom, ICC, and Globecom. He is a senior member of the IEEE and a member of the ACM.