Available online at www.sciencedirect.com
ScienceDirect
J. Parallel Distrib. Comput. 67 (2007) 715–726
Journal of Parallel and Distributed Computing
www.elsevier.com/locate/jpdc

QoS and data relaying for wireless sensor networks

Sylvia Tai, Robert R. Benkoczi*, Hossam Hassanein, Selim G. Akl

School of Computing, Queen's University, Kingston, Ont., Canada K7L 3N6

Received 6 April 2006; received in revised form 20 December 2006; accepted 27 January 2007
Available online 4 March 2007

Abstract

In this paper we study the effects of data relaying in wireless sensor networks (WSNets) under QoS constraints with two different strategies. In the first, data packets originating from the same source are sent to the base station possibly along several different paths, while in the second, exactly one path is used for this purpose. The two strategies correspond to splitting and not splitting relaying traffic, respectively. We model a sensor network architecture based on a three-tier hierarchy of nodes which generalizes to a two-tier WSNet with multiple sinks. Our results apply therefore to both types of networks. Based on the assumptions in our model, we describe several methods for computing relaying paths that are optimal with respect to energy consumption and satisfy QoS requirements expressed by the delay with which data are delivered to the base station(s). We then use our algorithms to perform an empirical analysis that quantifies the performance gains and losses of the splittable and unsplittable traffic allocation strategies for WSNets with delay-constrained traffic. Our experiments show that splitting traffic does not provide a significant advantage in energy consumption, but can afford strategies for relaying data with a lower delay penalty when using a model based on soft-delay constraints.

Keywords: Sensor networks; Data relaying; Delay constraints

1. Introduction

A wireless sensor network (WSNet) consists of many small sensor devices that measure certain properties of the environment in which they are deployed. The sensors (SN) typically communicate by radio signals and co-operate to transmit their measurements to one or more base stations [3]. Many mission-critical applications of WSNets generate traffic that have a stringent delay requirement. While it is important to provide a timely delivery of data for these applications, an efficient use of the network's limited energy resource must also be considered. Sensor nodes typically operate on batteries and have finite energy, but in many applications, the network is expected to have a long operating lifetime. Compared to sensing and data processing, data communication typically incurs the highest energy consumption [3]. Therefore, the task of coordinating data communications in the network to deliver delay-constrained traffic while also making efficient use of the limited energy resource, is an important topic for research.

1.1. The WSNet model

The sensor network architecture we consider in this paper is hierarchical and consists of three types of nodes, SN, relay nodes (RN), and relay gateways (RG). The SN create and communicate data. The RN receive data packets from SN and retransmit them to other RN or to the RG. The RG are capable of direct communication with the base station. The RG can also be viewed as a collection of multiple base stations, each with different capabilities.

Hierarchical architectures for WSNets were proposed earlier in the literature [10,19,20]. Most existing hierarchical architectures employ two classes of nodes, SN which generate data, and RN which are placed to insure the connectivity of SN when their data transmission range is too small. Even when the radio transmitters of the SN are powerful enough to reach the base station directly, the addition of RN could improve energy consumption and the quality of service by reducing the transmission

* Corresponding author. Fax: +1 613 533 6513.
E-mail addresses: tai@cs.queensu.ca (S. Tai), rbenkocz@cs.queensu.ca (R.R. Benkoczi), hossam@cs.queensu.ca (H. Hassanein), akl@cs.queensu.ca (S.G. Akl).

0743-7315/$ - see front matter
doi:10.1016/j.jpdc.2007.01.009

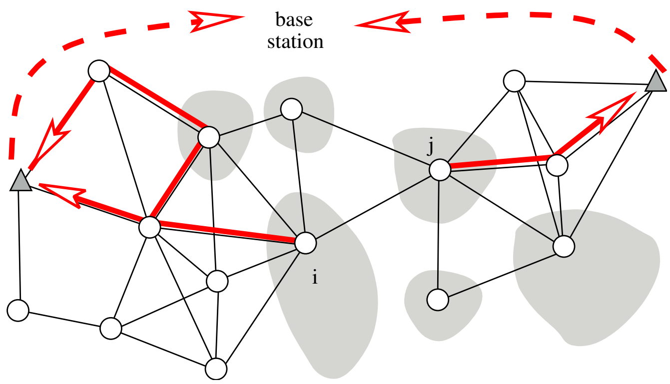


Fig. 1. An illustration of the three-tier architecture. The SN are represented as gray regions in the plane; the RN are represented as empty circles and the RG as gray triangles; some RN collect data from SN and also relay data on behalf of other RN. The data traffic of node i is split and that of node j is not split.

range and thus the interference. Naturally, as the number of RN increases, data delay caused by relaying also increases and RN closer to the base station consume more energy because they handle more traffic.

The three-layer hierarchical architecture is designed to address the conflicting issues between energy efficiency and QoS expressed as the delay in delivering data to the base station. In our model, energy requirements are translated into capacities assigned to the nodes in the network (see Section 2). Data delay, the time period between the moment the information is captured by sensor nodes and the moment it is delivered to the base station, depends on the length of the relaying path along which data flows. The problem is to select, for each RN, one or more paths to one or more RG along which information from the RN is relayed in such a way that the energy consumed by the system is as small as possible and the traffic does not exceed the node capacities (Fig. 1).

In previous work on two-tier WSNet, Tang et al. [17] considers the placement of RN to insure the connectivity of the network. Mhatre et al. [12] studies the node placement problem to minimize the cost of the network and to guarantee a certain lifetime. Xu et al. [20] and Wang et al. [18,19] consider computing the placement of RN given that SN are already placed when RN have unbounded and limited energy sources, respectively.

Clustering is a related technique that has been used mainly in ad hoc wireless networks, but also in WSNet routing protocols, with the goal of reducing the energy consumption of the system [6,14]. With clustering, a subset of the nodes that participate in a WSNet is automatically selected to perform special duties which require more energy than usual (sending data to the base station, performing calculations on the data, etc.). These nodes are the cluster heads. Several protocols were designed for the selection of the cluster heads in the context of ad hoc wireless networks (among many other papers we cite [4,5]). For sensor networks, the most widely known protocols are LEACH [10] and HEED [21].

QoS routing protocols in sensor networks have focused on timeliness and reliability of data. The distinguishing feature of these protocols, as compared to similar protocols for ad hoc wireless networks, is that they also try to be energy efficient. Examples are sequential assignment routing [15], SPEED [9] and weighted fair queuing [1]. A survey of routing protocols in wireless networks is given in [2].

However, to the best of our knowledge, no study has been conducted yet to assess the impact of a splittable or unsplittable traffic allocation strategy in the context of WSNets with delay constraints. In our paper, we perform an empirical analysis to quantify the performance gains and losses of a splittable and unsplittable traffic allocation strategy for WSNets with delay-constrained traffic. The study offers unique insights into the effects of splitting traffic on the delay and energy consumption behavior of WSNet applications with delay requirements. We observe that when traffic is delay constrained, splitting traffic does not provide a significant advantage in energy consumption, but can result in a lower data delay penalty.

2. Problem setup: models and assumptions

Topology: The WSNet is modeled as a general directed graph $D = (N, A)$ with node set N and arc set A. N is the union of the set of RN R and RG G. The SN are not explicitly modeled in the network topology. Their contribution is accounted by the amount of data traffic they generate. All RNs are assumed to transmit data using a fixed transmission range (i.e. RNs do not have dynamic power control). An arc $(i, j) \in A$, if node i can transmit directly to node j, for $i \in R$ and $j \in N$.

Data traffic: Data traffic is associated with the RNs and the arcs of D. We refer to the total amount of data that a RN i receives directly from the SNs, as being *generated* by RN i, and denote it by $\beta(i)$. The traffic associated with an arc $(i, j) \in D$ represents the data sent from RN i to a RN (or RG) j. The traffic flow on the arcs of D is associated with the RN that generated the traffic. We denote by $x^r(i, j)$ the amount of data generated by RN r that is relayed from RN i to RN (or RG) j.

Communication delay: A constraint on the traffic delay is modeled by imposing a limit on the length of the relay paths. A hop limit is associated with each RN i, denoted H^i. The data generated by RN i must be delivered on paths with no more than H^i hops in order to satisfy the traffic delay constraint. We refer to a path that exceeds the hop limit as a *delay-infeasible* path, and a hop in excess of the hop limit as a *violated hop*.

Energy supply: RNs and RGs have finite energy. Using the assumption that RNs transmit data using a fixed transmission range, the total amount of data that a RN is capable of relaying with its given energy supply can be derived [19]. We refer to this traffic amount as the *capacity* of a node i, and denote it by $\gamma(i)$. Capacity constraints are associated with each RN (resp., RG) and represent an upper bound on the total amount of data the node can relay with its given energy supply.

Energy consumption in communication: The energy consumed from communication in the network is modeled using a non-negative cost factor that is associated with the arcs of D. The cost of arc (i, j), denoted c_{ij}, represents the cost of relaying one unit of data from node i to node j. The cost c_{ij} includes (1) the cost of receiving one unit of data at node i, and (2) the cost of transmitting one unit of data from node i to node j. The energy consumption model used in [11] can be used to derive the cost factor associated with the arcs of D.

Other useful notations: Given node i, we denote the set of all arcs entering i by $\delta^-(i)$, and all arcs leaving i by $\delta^+(i)$,

$$\delta^-(i) = \{(j,i) | (j,i) \in A\}, \tag{1}$$

$$\delta^+(i) = \{(i,j) | (i,j) \in A\}. \tag{2}$$

If S represents a subset of the arcs of the graph, then we use $x(S)$ to represent the sum of values of function x over the elements of S, $x(S) = \sum_{a \in S} x(a)$.

We propose two different problems of selecting relaying paths for transmission with QoS constraints, one that splits the traffic from a source, and another that does not split the traffic. We call the first problem *Constrained Splittable Flow Allocation* problem (CSFA) and the second *Constrained Un-splittable Flow Allocation* problem (CUFA). In the following sections we describe each problem and discuss several linear and integer programming models for solving them.

3. The CSFA problem with hard delay constraints

In this problem, we must compute paths for every data packet so that the total number of intermediary nodes the packet is relayed through before it reaches a RG is not larger than a given value. There are two possible ways to formulate the delay constraint, one using flow functions as for the previous problem, and another using feasible paths. Similar models have been previously used in the context of network design problems [8].

3.1. Problem formulations

Flow functions: Consider a family of flow functions $x^r : A \to \mathbb{N}$ for all RN r. Function x^r represents the flow of data collected by RN r from sensor nodes. Let H^r be the upper bound on the length of any relaying path for the data generated at node r. To count the number of intermediary nodes that the flow originating at some RN r traverses, we assign an index to flow x^r that represents the hop count of the data represented by that flow function. Whenever a vertex relays the flow of data, this index is incremented. If we denote this index by h, then our problem becomes to compute a family of functions $x_h^r : A \to \mathbb{N}$, for all $r \in R$ and $0 \leqslant h \leqslant H^r$, given that certain constraints are satisfied.

These constraints are the following. Flow conservation relations,

$$x_0^r(\delta^+(r)) = \beta(r) \quad \forall r \in R, \tag{3}$$

$$x_{h+1}^r(\delta^+(i)) - x_h^r(\delta^-(i)) = 0$$
$$\forall r \in R, \quad \forall i \in R, \quad i \neq r, \quad 0 \leqslant h \leqslant H^r - 1. \tag{4}$$

Eq. (3) insures that a total flow with index zero is generated at every RN and equals the demand of that node. Eq. (4) states that every RN simply relays the information it receives and increments the hop count. RG do not have flow constraints because they act as sinks for the flow. The only constraints associated with RG are capacity constraints. The capacity constraints are

$$\sum_{r \in R} \sum_{h=0}^{H^r} x_h^r(\delta^-(i)) + \beta(i) \leqslant \gamma(i) \quad \forall i \in G \cup R, \tag{5}$$

where $\beta(k) = 0$ for all $k \in G$.

The delay constraints are

$$x_{H^r}^r(\delta^-(i)) = 0 \quad \forall i \in R. \tag{6}$$

The relation makes sure that no RN can receive a flow of data that has reached the maximum delay threshold of H^r.

Feasible paths: This formulation is perhaps more direct than the one using flow functions because we assign flow of data to paths from RN to RG in the graph and not just to the arcs of the graph. Let r be a RN and k a RG in the description that follows. Denote by $\mathcal{P}(r)$ the set of all directed paths in D, starting at r and ending at a RG, with length no more than H^r. Similarly, let $\mathcal{P}(k)$ be the set of path from any RN r to the given RG k, and let $\mathcal{P}(r,k) = \mathcal{P}(r) \cap \mathcal{P}(k)$, i.e. the set of feasible paths from r to k. Let $\mathcal{P}$ be the set of all paths of feasible length ending at a RG. We use index s to denote paths in the graph and we define function $x : \mathcal{P} \to \mathbb{N}$ that returns the value of data flow through a given path.

The delay constraints are now implicit; they are captured in the structure of set $\mathcal{P}$. The only constraints needed are those that insure the demand of every RN is met (7) and that the capacity of every node is not violated (8),

$$\sum_{s \in \mathcal{P}(i)} x(s) = \beta(i) \quad \forall i \in R, \tag{7}$$

$$\sum_{\substack{s \in \mathcal{P} \setminus \mathcal{P}(i) \\ i \in s}} x(s) + \beta(i) \leqslant \gamma(i) \quad \forall i \in G \cup R. \tag{8}$$

3.2. Solution for problem CSFA

The flow functions formulation (3)–(6) represent a linear program that can be used directly to obtain the optimal solution to the problem. The disadvantage of this approach is the size of the problem involved. The number of constraints needed is $O(|R|^2 \max_{r \in R} H^r)$ which is too large. If we wish to solve problem CSFA for large scale graphs, we need to work with formulations involving a number of constraints that are linear in the size of the graph (preferably linear in the number of nodes and independent of the number of edges). The feasible paths formulation has this property even though the number of variables is exponential. We can handle this using *column generation*.

Column generation is an implementation of the simplex algorithm for solving linear programs where the non-basic variables are not stored in memory but are generated by solving

- Initial phase: find a set of initial columns C_{init}. Construct an empty master problem.
- Let $c \leftarrow C_{init}$
- REPEAT
  - · Add c to master problem
  - · Solve master LP problem to optimality; get the dual_optimal_variables
  - · Let $c \leftarrow$ GenCol(dual_optimal_variables)

  UNTIL c has non-negative reduced cost
- return optimal solution of master problem

Fig. 2. Main algorithm for solving LPs by column generation.

- assign a weight $w((i,j)) \leftarrow c((i,j)) + y(j)$ for all edges (i,j)
- FOR all $r \in R$ DO
  - · Compute the shortest path tree from source r in graph D using edge weights w
  - · select the RG closest to r whose shortest path $s_{shortest}$ has fewer than Θ links; let $d_{shortest}$ be the distance with edge weights w
  - · Let $rc_r \leftarrow d_{shortest} - z(r)$
- Select node r and the corresponding path s_{opt} with minimum rc_r over all $r \in R$; return the column of s_{opt}.

Fig. 3. Column generation algorithm: GenCol.

a column generation sub-problem. Here by column of a variable we refer to the actual column in the constraint matrix for the variable. As described in [7], the simplex algorithm identifies a non-basic variable whose reduced cost is negative (i.e. who do not satisfy feasibility conditions of the dual problem) and then makes this variable basic by a process called pivoting. With column generation, the non-basic variables are not stored in the memory because their number is too large (for example, exponential as in our case). The column generation procedure consists of two processes, (a) a master LP problem and (b) a column generation sub-problem. The master problem contains a subset of the variables of the full formulation of the problem we are solving, but the dual of the master problem contains all the dual variables of the full problem. The optimal solution of the master problem might coincide with the optimal solution of the full problem if, by chance, the non-zero variables in the optimal solution of the full problem happen to be part of the master. If not, the sub-problem identifies another non-basic primal variable with a negative reduced cost (violating the dual feasibility condition) by constructing it. If no variable with a negative reduced cost can be constructed, then all dual feasibility conditions are satisfied and the optimal solution of the master problem coincides with that of the full problem. Otherwise, the new variable is added to the master problem, a new optimum solution is found using off the shelf LP software to solve the master and the new set of dual variables are again used as input for the column generation sub-problem. The whole procedure is sketched in Fig. 2.

We are interested in the LP dual of problem (7)–(8). We associate dual variable $z(i)$ with constraint (7) and dual variable $y(i)$ with inequality (8). Variable $z(i)$ is defined for all $i \in R$ and variable $y(i)$ for all $i \in R \cup G$ and it must be positive. The objective function we want to optimize in the primal problem (7)–(8) is the total weighted length of the paths selected to receive flow. Let $l(s)$ denote the length of path s, i.e. $l(s) = \sum_{(i,j)\in s} c((i,j))$, where c is the cost function for the arcs. The objective of the primal problem is then

$$\min \sum_{s\in\mathcal{P}} l(s)\cdot x(s) \quad \text{s.t. (7), (8) and } x(s)\geqslant 0. \tag{9}$$

The LP dual of (9) is

$$\max \sum_{i\in R} \beta(i)z(i) - \sum_{i\in G\cup R} \big(\gamma(i) - t\beta(i)\big)y(i)$$

$$\text{s.t. } z(r) - \sum_{j=1}^{m} y(i_j) \leqslant l(s) \quad \forall s\in\mathcal{P}, \quad s=(r,i_1,i_2,\dots,i_m),$$

$$y(i)\geqslant 0 \quad \forall i\in G\cup R. \tag{10}$$

The reduced cost of any path $s=(r,i_1,i_2,\dots,i_m)$ from the primal problem is

$$rc_s = l(s) - z(r) + \sum_{j=1}^{m} y(i_j)$$

$$= \sum_{j=1}^{m} \big(c((i_{j-1},i_j)) + y_j\big) - z(r), \tag{11}$$

where $i_0 = r$. From this equation, we can design a very simple procedure for the generation of columns (see Fig. 3). Since we want to construct a path where the reduced cost expression is minimized, we observe that we only need to compute shortest paths in graph D. This procedure requires solving the shortest path using edge weight $w((i,j)) = c((i,j)) + y(j)$. The edge weights are all positive.

The only issue that needs to be explained is the initial phase of the main algorithm. How do we initialize the master problem? There are two ways. One is to use any kind of heuristic algorithm that returns a feasible solution to the delay-constrained multi-path relaying problem. Then, we simply use the columns corresponding to the paths carrying flow as the initial columns in the master problem. The second is to use artificial variables in another LP problem very similar to (7)–(8).

We associate artificial variable $v(i)$ for every constraint (7) and define the following LP problem:

$$\min \sum_{i\in R} v(i) \text{ s.t. (8) and } v(i) + \sum_{s\in\mathcal{P}(i)} x(s) = \beta(i). \tag{12}$$

Here again, variables $x(s)$ are generated using column generation, but we can start the master problem by adding the artificial variables and considering all other $x(s)$ equal to zero. The column generation procedure is identical to that in Fig. 3 except that the weight of an edge (i,j) is set to $w((i,j)) = y(j)$. The column generation algorithm will stop as soon as the optimal solution of the current master problem reaches zero. This means that all artificial variables are null and the columns from the master problem define a feasible solution to the delay-constrained multi-path relaying problem. If, on the other hand, the optimal solution to (12) is positive, then problem CSFA is infeasible. This completes the description of the algorithm.

We stress that the algorithm described above, based on solving a linear program with the simplex algorithm with column generation, is optimal. Although the simplex algorithm is a heuristic, it is very efficient in practice.

4. Soft-delay constraints and the constrained un-splittable and splittable flow allocation problems (CUFA and CSFA)

When modeling problem CUFA as an integer program, we can choose a more compact constraint for the delay limit, without requiring to index the flow with the hop count as in the previous section. As a result, the number of variables and constraints can be reduced considerably. For the problem sizes we used in our experiments, this allows us to use off-the-shelf IP solvers since we do not need to consider a column generation procedure. Of course, to solve problem sizes that are much larger, we can still use column generation. We comment on this approach at the end of this section.

To formulate CUFA as an IP problem, we define an augmented graph $D' = (V', A')$, where $V' = V \cup \{v\}$ and $A' = A \cup \{\bigcup_{g \in G}(g, v)\}$. Node v represents the sink of all traffic generated in the network and $c_{iv} = 0, \forall\, i \in G$. We define a normalized flow function $y^r : A \to \{0, 1\}$ and consider $x^r(i, j) = \beta(r) \cdot y^r(i, j)$. If $y^r(i, j) = 1$, then the entire traffic generated by RN r flows through arc (i, j).

4.1. Soft-delay constraints for CUFA and CSFA

Using the normalized flow functions, we can easily relax the delay constraints, defining a new version of problems for both CUFA and CSFA with *soft-delay constraints*. In an empirical study, the advantage of using soft-delay constraints is that it gives an indication of the difficulty of providing the QoS based on delay.

If there are no paths within the hop limit of H^i that can be used to deliver the traffic generated by RN i, then we relax the hop count requirement to permit paths with J hops in the solution, for some $J > H^i$. We introduce a penalty term in the formulation that increases the value of the objective function when a delay constraint is violated. The objective to minimize is the sum of the total energy spent on communication and the penalty incurred from the violation of delay constraints.

Let $z(i)$ be the number of hops in excess of the hop limit H^i, that is used on a path to deliver traffic generated by RN i. The penalty for the violation of delay constraints is expressed as a function of $z(i)$. To reduce excess delay in data delivery, the penalty function f should discourage the selection of long delay-infeasible paths (i.e. the penalty incurred for data delivered on each additional violated hop on a delay-infeasible path should be non-decreasing). Note that the choice of f affects the selection of relay paths used in the optimal solution. For example, using an exponential function for f will penalize the selection of long delay-infeasible paths more severely than using a linear function for f. Here, we define f as a step function to incorporate different tolerances for the delayed delivery of data. We define L penalty levels, where each penalty level corresponds to a given magnitude of delay violation. The penalty levels create intervals of delay violation $I_1, \ldots, I_L$, as shown in Fig. 4.

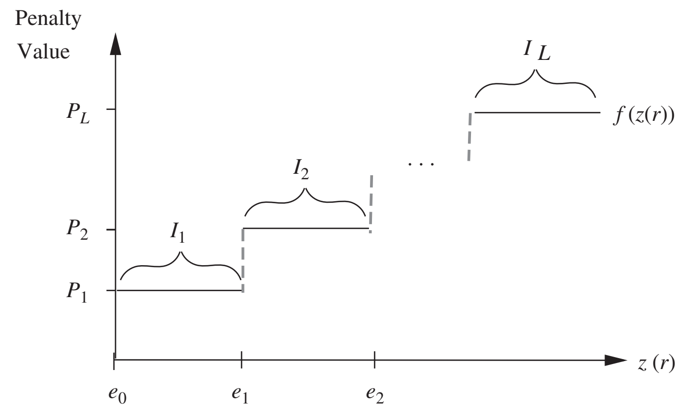


Fig. 4. The penalty for delay violation is modeled as a step-function $f(z(r))$ of the number of hops that exceed the hop limit. Each interval $I_1, \ldots, I_L$ of delay violation is associated a penalty value.

Let $(e_{i-1}, e_i]$ be the interval I_i, for $i = 1, \ldots, L$. Let l_i denote the length of interval I_i. Each interval I_i is assigned a penalty value P_i, where $P_{i-1} \leqslant P_i$, for $i = 2, \ldots, L$. P_i represents the amount of increase in the objective value, per unit of traffic that is delivered on the kth violated hop of a delay-infeasible path, for some $k \in I_i$. For example, we can define three penalty levels, with $P_1 = 50$, $P_2 = 100$, and $P_3 = 200$ for the intervals $(0, 1]$, $(1, 3]$, and $(3, \infty]$, respectively. A path carrying 2 units of traffic that exceeded the hop limit by 4 hops will incur a penalty of $2 \cdot (1 \cdot P_1 + 2 \cdot P_2 + 1 \cdot P_3) = 2 \cdot (50 + 200 + 200) = 900$. For each RN r, we introduce non-negative integer variables $z_1(r), z_2(r), \ldots, z_L(r)$ corresponding to the L penalty levels.

The IP formulation for CUFA is as follows:

- Objective function:

$$\min \left\{ \sum_{a \in A'} c(a) \cdot \left(\sum_{r \in R} \beta(r) y^r(a) \right) + \sum_{r \in R} \beta(r) \sum_{i=1}^{L} P_i z(r)_i \right\}.$$

- Flow conservation constraints:

$$y^r\left(\delta^+(i)\right) - y^r\left(\delta^-(i)\right) = \begin{cases} 1 & \text{if } i = r, \\ -1 & \text{if } i = v, \\ 0 & \text{otherwise,} \end{cases} \quad \forall\, r \in R, \ i \in R \cup G. \tag{13}$$

- Node capacity constraints:

$$\sum_{r \in R \setminus \{i\}} \beta(r) \cdot y^r\left(\delta^-(i)\right) + \beta(i) \leqslant \gamma(i) \quad \forall\, i \in G \cup R. \tag{14}$$

- Delay constraints:

$$\sum_{a\in A'} y^r(a) - \sum_{i=1}^{L} z_i(r) \leqslant H^r + 1 \quad \forall\, r \in R, \tag{15}$$

$$z_i(r) \leqslant l_i \quad (i = 1, \ldots, L) \quad \forall\, r \in R. \tag{16}$$

Problem CSFA with soft-delay constraints can be formulated using the same set of constraints and objective function as the feasible paths formulation for problem CSFA with hard delay constraints in the previous section. However, a different column generation sub-problem is needed to generate new variables to add to the restricted master problem. In the formulation, each variable $y(r)$ represents a relay path in the network with origin $r \in R$. The value of $y(r)$ represents the fraction of total traffic generated by $r \in R$ that is sent on the path represented by $y(r)$. The structure of the column generation sub-problem is unchanged; it is still a shortest path problem on digraph D, but with a set of modified arc costs that adjusts for the penalty incurred from using violated hops to deliver traffic in the network. The sub-problem can be solved using a variation of Dijkstra's algorithm. The coefficient of $y(r)$ in the objective function represents the sum of the cost of transmitting one unit of traffic on the path represented by $y(r)$, and the penalty incurred from transmitting one unit of traffic on each violated hop on that path.

The converse is however not true. Problem CSFA with hard delay constraints cannot be modeled with the compact delay formulation based on the normalized flow. The compact delay constraint formulation imposes a limit on the sum of the fractional normalized flow values. Although the sum may be less than the hop limit, there may exist individual paths with lengths greater than the hop limit but with a small traffic assignment.

Note that for both problems CSFA and CUFA, a relaying strategy is given implicitly in a feasible solution to the LP/IP formulation. The assignment of values to variables obtained in an optimal solution (i.e. a feasible solution with the best objective value) to the formulation indicates a set of relay paths and the traffic amount sent on each path so that the sum of the total energy spent on communication and the penalty incurred from the violation of delay constraints is a minimum.

5. Experiment setup and results

Experiments are performed to study how network performance may differ under the choice of a splittable and unsplittable flow allocation strategy for WSNets when traffic is delay constrained. This section describes the experiment setup and our methodology.

The algorithms for solving problems CSFA and CUFA solve the LP and IP formulations described in Section 2. The algorithms are implemented in C/C++, using MOSEK [13] as the underlying optimization solver with the default configurations. In the experiment, we assume that all RNs transmit data using the same fixed transmission range. The low production cost of RNs with no power control and the convenience of a homogeneous deployment may justify the practicality of this assumption. However, a network of RNs with fixed but heterogeneous transmission ranges can simply be represented by using different energy cost coefficients in the objective function of the LP/IP formulation. All networks in our problem instances are assumed to have a lifetime requirement of 1000 time units. Lifetime requirement, in this context, is a given length of time that the network is required to remain in operation to collect and deliver sensor data to the base station(s). Given the sensing rate of the sensor nodes and a lifetime requirement, the total amount of traffic generated at each RN for the duration of the lifetime requirement can be derived. The energy cost of relaying one unit of data is derived using the energy consumption model in [11] and the radio characteristics of the RNs given in Table 1.

Table 1
Radio characteristics of RNs

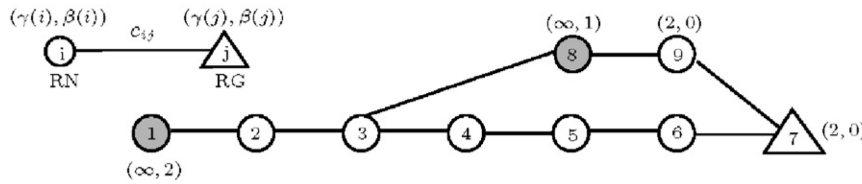
Radio function	Energy dissipation
Transmission (α)	50 nJ/bit
Reception (ρ)	50 nJ/bit
Transmit amplifier (β)	0.0013 pJ/bit/m^4 (for path loss index 4)

All topologies used in the experiment are generated randomly and form a connected network in the plane. The procedure for generating the random topologies is given in [16]. Here, we assume that the communication links are bidirectional. The sensor field dimension is 500 m × 500 m, and the transmission radius is 5 m for all RNs.

We consider a scenario where the sensor data received by the RNs deployed at different locations of the sensor field are equally time-critical. Hence, all RNs are assumed to have the same delay requirements and must deliver data on relay paths that are no longer than H hops in order to satisfy the delay constraint. In the experiment, we choose a linear penalty function. i.e. the penalty function has one penalty level P_1 for the interval $(0, \infty]$. The penalty function penalizes all delay constraint violations equally with an increase of P_1 units in the objective value per unit of traffic that is delivered on each violated hop, regardless of the length of an individual delay-infeasible path. For example, consider a problem instance with hop limit $H = 3$. A solution in which a RN delivers 3 units of traffic on a path of length 5 and a solution in which a RN delivers 2 units of traffic on a path of length 6 and 1 unit of traffic on a path of length 3 will have the same penalty. Thus, the chosen penalty function tries to minimize the total amount of traffic that is delivered on violated hops in the optimal solution. We can consider the penalty term in the objective function as the value P_1 multiplied by the *penalized traffic* (PT). The PT is the sum of the traffic amount delivered on each violated hop, over all violated hops used in the optimal solution. The penalty value P_1 is chosen to be a very large value to ensure that a solution with one or more delay constraint violations is less preferable than any solution without a delay constraint violation. P_1 is set to $|R| \cdot H \cdot E$, where E is the energy cost of relaying one unit of traffic.

Table 2
Size of the randomly generated networks for experiment 3

Case	RN	RG	No. links	Neighbors (Avg. per RN)	Aggregate capacity (kb)	
					RN	RG
1	50	3	928	17	465,000	454,975
2	50	4	973	18	483,000	340,635
3	50	5	630	11	559,000	645,916



a UUFA, Objective value =14

RN	Flow Amount	Path
1	2	$1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7$
8	1	$8 \to 9 \to 7$

b CUFA, Objective value =15

RN	Flow Amount	Path
1	2	$1 \to 2 \to 3 \to 8 \to 9 \to 7$
8	1	$8 \to 3 \to 4 \to 5 \to 6 \to 7$

c USFA, Objective value =13

RN	Flow Amount	Path
1	1	$1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 7$
1	1	$1 \to 2 \to 3 \to 8 \to 9 \to 7$
8	1	$8 \to 9 \to 7$

d CSFA, Objective value =15

RN	Flow Amount	Path
1	2	$1 \to 2 \to 3 \to 8 \to 9 \to 7$
8	1	$8 \to 3 \to 4 \to 5 \to 6 \to 7$

Fig. 5. Example of a problem instance where the presence of delay constraints given by hop limit $H = 5$, results in an optimal solution that has a higher total energy consumption than without the delay constraints. Unless specified, all RNs generate zero traffic demand and have infinite capacity. The energy cost of delivering one unit of traffic per hop is 1.

In the experiment, we compare the total energy consumption, PT, and amount of delayed traffic (DT) in the optimal solution to CSFA and CUFA over problem instances defined on the same topology, but with a variable aggregate traffic load. The amount of DT refers to the total amount of traffic delivered on delay-infeasible paths in a given solution. For example, the optimal solution to problem CUFA (with hop limit $H = 4$) shown in Fig. 7(b) has 4 units of PT and 2 units of DT. Aggregate traffic load T_{total} is the total amount of traffic generated by all RNs in the network for the duration of the required lifetime. In the experiment, a RN can generate either a *high* traffic load or a *low* traffic load, corresponding to a data rate of 5 and 1 kb per time unit, respectively. In practice, sensor nodes can be deployed according to different deployment densities and traffic conditions in the network can change in real-time (e.g. due to an increase/decrease in the sensing rate of nodes in areas that detected a high/low interest in the sensed phenomena). Consequently, the amount of traffic load generated at each RN may be different. RNs are assigned random capacities in the range of 5000–15,000 kb. We define the aggregate RG (resp., RN) capacity as the sum of the capacities of all RGs (resp., RNs) in the network. Each RG is assigned a random capacity in the range of 5000 kb to $T_{\max}$, where $T_{\max}$ is the maximum aggregate traffic load (i.e. when all RNs generate a high traffic load) that can be generated in the network for the duration of the required lifetime.

For the same topology, the percentage of RNs carrying a high traffic load is varied from 0% to 100%, in increments of 10%. For each percentage value chosen, the results are averaged over 10 problem instances with different random assignments of high/low traffic loads to RNs. The experiment is performed for the three randomly generated networks listed in Table 2 with 50 RNs, and the number of RGs ranging from 3 to 5, in increments of 1. The experiment is repeated using the same problem instances for hop limit $H = 2$, 3, and 4.

5.1. Experiment results

We refer to the splittable and unsplittable traffic allocation problems without delay constraints as problems USFA and UUFA, respectively. Problems CSFA and CUFA can be viewed as a special case of problems USFA and UUFA, i.e. the total energy consumption in the optimal solution to USFA (resp., UUFA) is a lower bound on the total energy consumption in the optimal solution to CSFA (resp., CUFA). The presence of delay constraints can increase the total energy consumption for both problems, as shown in Fig. 5. Fig. 5 gives an example of a

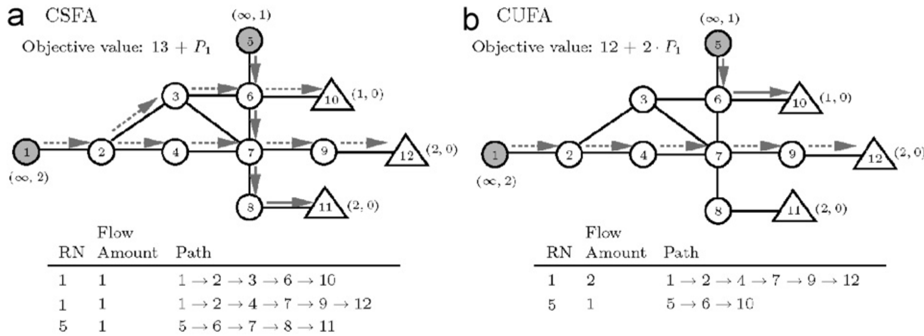


Fig. 6. Example of a problem instance with hop limit $H = 4$ where the total energy consumption in the optimal solution to CSFA is *greater* than that of CUFA. A RN (circle) or RG (triangle) is labeled with the pair (capacity, traffic load). Unless specified, all nodes generate zero traffic load and have infinite capacity. The energy cost of delivering one unit of traffic per hop is 1. The optimal solution to CSFA shown in (a) has an energy cost of 13, while the optimal solution to CUFA shown in (b) has an energy cost of 12.

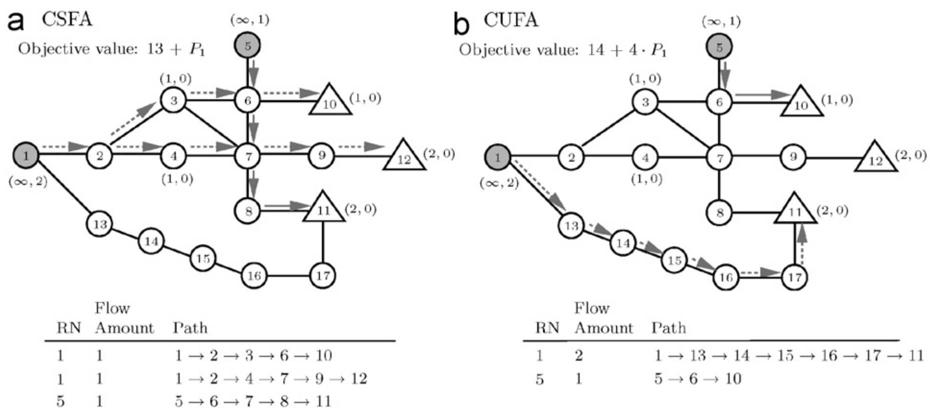


Fig. 7. Example of a problem instance with hop limit $H = 4$ where the total energy consumption in the optimal solution to CSFA is *less* than that of CUFA. A RN (circle) or RG (triangle) is labeled with the pair (capacity, traffic load). Unless specified, all nodes generate zero traffic load and have infinite capacity. The energy cost of delivering one unit of traffic per hop is 1. The optimal solution to CSFA shown in (a) has an energy cost of 13, while the optimal solution to CUFA shown in (b) has an energy cost of 14.

problem instance in which the minimum total energy consumption required to deliver unsplittable (resp., splittable) traffic having the least delay constraint violations (given by hop limit $H = 4$), as shown in (b) (resp., (d)), is greater than the minimum total energy consumption required to deliver unsplittable (resp., splittable) traffic when no delay constraints are present, as shown in (a) (resp., (c)).

The total energy consumption in the optimal solution to USFA is a lower bound on the total energy consumption in the optimal solution to UUFA [16]. However, when soft-delay constraints and penalties are used in the formulation, the energy consumption in the optimal solution to CSFA can be greater than or less than that of CUFA, as shown in Figs. 6 and 7, respectively. The examples illustrate the possible effects on energy consumption due to differences in the ability of a splittable and unsplittable traffic allocation strategy to reduce the penalty from delay constraint violations.

Tables 3–5 show the additional total energy consumption (E^+), PT, and amount of DT in the optimal solution to CUFA compared to CSFA. Note that a problem instance that is infeasible for CSFA necessarily implies that it is also infeasible for CUFA. The results are computed over those problem instances where both CSFA and CUFA have an optimal solution. The results are shown only for those levels of aggregate traffic load where there is at least one feasible solution to CUFA out of the 10 random problem instances. Although a positive or negative valued E^+ is theoretically possible, a positive valued E^+ that is consistently observed for all three networks indicate that as aggregate traffic load increases, on average, the total energy consumption in the optimal solution to CUFA is greater than the total energy consumption in the optimal solution to CSFA. In fact, less than 1% of the problem instances have a CUFA energy consumption that is less than CSFA. The average difference in total energy consumption is relatively small, ranging from 0% to 3.13%, 0% to 3.65%, and 0% to 3.47% for hop limits $H = 2$, 3, and 4, respectively. The largest observed difference in total energy consumption for a given problem instance is 7.45%. We also observe instances where the average E^+ is as small as 0 along initial increases in aggregate traffic load. For both problems CSFA and CUFA, RNs will favor the

Table 3
E^+: additional total energy consumption for CUFA compared to CSFA, expressed as a percentage of CSFA total energy consumption

Network size	High traffic RNs (%)	Aggregate traffic load (kb)	$H=2$		$H=3$		$H=4$	
			Avg.	Max.	Avg.	Max.	Avg.	Max.
50 RN	0	50,000	0	0	0	0	0	0
3 RG	10	70,000	0	0	0	0	0	0
	20	90,000	0	0	0	0	0	
	30	110,000	0.28	1.92	0.52	4.33	0.28	1.92
	40	130,000	0.63	3.85	0.59	4.27	0.63	3.85
	50	150,000	1.87	3.52	2.46	5.19	1.67	3.52
	60	170,000	2.03	5.01	2.38	7.14	2.38	7.14
50 RN	0	50,000	0	0	0	0	0	0
4 RG	10	70,000	0	0	0	0	0	0
	20	90,000	0.54	3.31	0.54	3.31	0.54	3.31
	30	110,000	1.03	5.24	1.04	2.62	0.77	2.62
	40	130,000	1.44	5.09	1.35	5.09	1.30	5.09
	50	150,000	2.31	4.15	2.26	4.15	2.31	4.15
	60	170,000	3.13	7.45	3.65	7.45	3.13	7.45
	70	190,000	0.68	0.68	0.68	0.68	0.68	0.68
50 RN	0	50,000	0	0	0	0	0	0
5 RG	10	70,000	0	0	0.08	0.76	0.60	5.17
	20	90,000	0.22	1.74	0.28	2.84	0.22	2.23
	30	110,000	0.29	2.18	0.64	2.77	0.73	3.36
	40	130,000	0.82	2.23	0.44	2.61	0.44	1.34
	50	150,000	1.38	3.05	2.55	5.08	2.38	6.03
	60	170,000	1.51	2.88	2.40	3.55	3.33	5.54
	70	190,000	2.88	3.41	3.32	3.99	3.47	4.28

Table 4
PT: additional amount of penalized traffic for CUFA compared to CSFA, expressed in units of 1000 kb

Network size	High traffic RNs (%)	Aggregate traffic load (kb)	$H=2$		$H=3$		$H=4$	
			Avg.	Max.	Avg.	Max.	Avg.	Max.
50 RN	0	50,000	0	0	0	0	No delay	
3 RG	10	70,000	0	0	0	0	violation	
	20	90,000	0	0	0	0		
	30	110,000	0.60	4.00	0.40	4.00		
	40	130,000	1.50	9.00	0.40	4.00		
	50	150,000	4.50	10.00	2.00	4.00		
	60	170,000	6.25	15.50	2.17	9.00		
50 RN	0	50,000	0	0	0	0	No delay	
4 RG	10	70,000	0	0	0	0	violation	
	20	90,000	0.80	5.00	0	0		
	30	110,000	1.40	5.00	0.50	5.00		
	40	130,000	2.90	11.00	2.10	8.00		
	50	150,000	5.60	10.00	3.20	10.00		
	60	170,000	8.57	21.00	5.57	14.00		
	70	190,000	2.00	2.00	0	0		
50 RN	0	50,000	0	0	0	0	0	0
5 RG	10	70,000	0	0	0	0	0	0
	20	90,000	0	0	0	0	0	0
	30	110,000	0.10	1.00	0	0	0	0
	40	130,000	0.43	3.00	0.43	3.00	0	0
	50	150,000	1.97	4.69	2.38	8.00	0.63	5.00
	60	170,000	5.44	10.69	4.29	8.69	1.00	5.00
	70	190,000	9.19	11.69	5.50	8.00	2.50	5.00

Table 5
DT: additional amount of delayed traffic for CUFA compared to CSFA, expressed as a percentage of the aggregate traffic load

Network size	High traffic RNs (%)	Aggregate traffic load (kb)	$H = 2$		$H = 3$		$H = 4$	
			Avg.	Max.	Avg.	Max.	Avg.	Max.
50 RN	0	50,000	0	0	0	0	No delay	
3 RG	10	70,000	0	0	0	0	violation	
	20	90,000	0	0	0	0		
	30	110,000	−0.09	1.82	0.36	3.64		
	40	130,000	0.23	6.90	0.31	3.08		
	50	150,000	1.33	3.33	1.33	2.67		
	60	170,000	1.57	5.88	0.78	2.35		
50 RN	0	50,000	0	0	0	0	No delay	
4 RG	10	70,000	0	0	0	0	violation	
	20	90,000	0.33	1.11	0	0		
	30	110,000	0.36	0.91	0.45	4.55		
	40	130,000	0.85	4.62	1.62	6.15		
	50	150,000	1.33	4.00	2.40	6.67		
	60	170,000	1.76	4.11	3.28	8.24		
	70	190,000	1.05	1.05	0	0		
50 RN	0	50,000	0	0	0	0	0	0
5 RG	10	70,000	0	0	0	0	0	0
	20	90,000	0	0	0	0	0	0
	30	110,000	0.09	0.91	0	0	0	0
	40	130,000	0	0	0.33	2.31	0	0
	50	150,000	0.28	2.00	0.75	2.00	0.42	3.33
	60	170,000	0.66	2.17	1.29	2.94	0.59	2.94
	70	190,000	−0.16	−0.16	1.58	1.58	1.32	2.63

choice of shortest paths to relay data in order to minimize both total energy consumption and the delay penalty. Since most of the RNs are generating a low traffic load when aggregate traffic load is low, RNs will likely have sufficient capacity to relay the entire traffic load of a RN over a single shortest path that will result in the least amount of PT and energy consumption possible. In this case, a small difference in energy consumption and delay penalty is expected between problems CSFA and CUFA. In fact, along initial increases in aggregate traffic load where the average E^+ is 0, the average PT is also 0. A positive valued PT indicates that the amount of PT is consistently greater for CUFA than for CSFA. As well, a larger average PT value can be observed at higher levels of aggregate traffic load. The larger difference in the amount of PT is due to more traffic being sent on the violated hops of delay-infeasible paths in problem CUFA, when RNs on the shorter paths do not have sufficient capacity to carry the entire traffic load generated by a RN. However, the capacity limited shortest paths that cannot be used to deliver entire traffic loads can be used to deliver fractions of that traffic, if it is splittable. Therefore, the results suggest that when traffic is delay constrained, splitting traffic does not provide a significant advantage in energy consumption, but can afford strategies for relaying data with a lower delay penalty.

A positive valued DT indicate that on average, the best strategy for relaying splittable traffic results in less DT than the best strategy for relaying unsplittable traffic. In theory, problems CSFA and CUFA are not guaranteed to have a unique optimal solution, i.e. more than one feasible solution having the same optimal objective value may exist for a given problem instance. Since different optimal solutions for the same problem instance correspond to different relaying strategies, the amount of DT given by each strategy may be different. In the experiment, the DT for a given problem instance is computed from an optimal solution to CSFA and CUFA that is returned by the optimization solver. Table 5 shows that the average DT ranges from −0.09% to 1.76%, 0% to 3.28%, and 0% to 1.32% for hop limits $H = 2$, 3, and 4, respectively. The average absolute difference in DT ranges from 0% to 2.55%, 0% to 3.28%, and 0% to 1.32%, for hop limits $H = 2$, 3, and 4, respectively. Assuming that for the same problem instance, any optimal solution to CSFA (resp., CUFA) is as desirable as any other (since they have the same objective values), the results show that a small difference in the amount of DT is possible between the optimal strategies for splittable and unsplittable traffic.

6. Conclusions

In this paper, we presented a model based on linear and integer linear programming for finding an optimal allocation of splittable and unsplittable traffic in a WSNet, in which traffic is subject to hard and soft-delay constraints. Based on this model, we performed an empirical analysis to quantify the effects of splitting traffic in a WSNet with soft-delay-constrained traffic.

Experiment results show that splitting traffic does not provide a significant advantage in energy consumption, but can afford strategies for relaying data with a lower delay penalty when the capacity of the network is used to the limit. When the network capacity is in excess, there are no significant differences between the two relaying strategies. This also indicates that in applications where there is a mix of time sensitive and time insensitive data, we do not expect to observe significant differences between the two relaying strategies.

There is an interesting and surprising interpretation of our experiments. If we consider a network of RN without power control, we can view routing as being a relaying strategy. The relaying strategy splits the data if different relaying paths are used during the lifetime of the network. We assume that for the application at hand we have access to the expected amount of data traffic generated by the sensor nodes but we cannot predict the exact network conditions at a given moment in time. Naturally, the sensor network would adapt to the current conditions and would chose some relaying path for the data injected in the network to optimize a certain objective function. The path will change with time. If our only concern is energy conservation, our results indicate that one should consider precomputing a set of fixed paths to be used for relaying data during the lifetime of the network as an alternative to a more complex system that changes the relaying paths with the network conditions. On the other hand, if data delay is also important, one cannot afford to use a fixed path relaying strategy. In conclusion, we would like to point out that perhaps more care should be given to analyzing the particularities of WSNet applications since there are situations when routing can be replaced by a simple relaying strategy over fixed paths.

References

[1] K. Akkaya, M. Younis, Energy-aware delay-constrained routing in wireless sensor networks, Internat. J. Commun. Syst. 17 (2004) 663–687.

[2] K. Akkaya, M. Younis, A survey of routing protocols in wireless sensor networks, Ad Hoc Networks 3 (3) (2005) 325–349.

[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE Commun. Mag. 40 (8) (2002) 102–114.

[4] S. Banerjee, S. Khuller, A clustering scheme for hierarchical control in multi-hop wireless networks, in: Proceedings of IEEE INFOCOM, 2001.

[5] S. Basagni, Distributed clustering algorithm for ad-hoc networks, in: Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN), 1999.

[6] R.V. Dyck, L. Miller, Distributed sensor processing over an ad-hoc wireless network: simulation framework and performance criteria, in: Proceedings of IEEE Milcom, McLean, VA, 2001.

[7] S. Gass, Linear Programming: Methods and Application, McGraw-Hill, New York, 1985.

[8] L. Gouveia, P. Patricio, A.F. de Sousa, R. Valadas, Mpls over wdm network design with packet level qos constraints based on ilp models, in: Proceedings of IEEE Infocom, 2003.

[9] T. He, J. Stankovic, C. Lu, T. Abdelzaher, Speed: a stateless protocol for real-time communication in sensor networks, in: International Conference on Distributed Computing Systems (ICDCS), Providence, RI, 2003.

[10] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocols for wireless microsensor networks, in: Proceedings of the 33rd Hawaii International Conference on System Sciences, vol. 8, 2000, pp. 8020–8029.

[11] Y. Hou, Y. Shi, H.D. Sherali, S.F. Midkiff, On energy provisioning and relay node placement for wireless sensor networks, IEEE Trans. Wireless Commun. 4 (5) (2005) 2579–2590.

[12] V. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, N. Shroff, A minimum cost heterogeneous sensor network with a lifetime constraint, IEEE Trans. Mobile Comput. 4 (1) (2005) 4–15.

[13] Mosek, ⟨http://www.mosek.com⟩, 2006.

[14] G. Pottie, Wireless sensor networks, in: Proceedings of IEEE Information Theory Workshop, Killarney, Ireland, 1998, pp. 139–140.

[15] K. Sohrabi, J. Gao, V. Ailawadhi, G. Pottie, Protocols for self-organization of a wireless sensor network, IEEE Personal Commun. 7 (5) (2000) 16–27.

[16] S. Tai, R. Benkoczi, H. Hassanein, S. Akl, A performance study of splittable and unsplittable traffic allocation in wireless sensor networks, in: Proceedings of IEEE International Conference on Communications, Istanbul, Turkey, 2006.

[17] J. Tang, B. Hao, A. Sen, Relay node placement in large scale wireless sensor networks, Comput. Commun. 29 (4) (2006) 490–501.

[18] Q. Wang, K. Xu, H. Hassanein, G. Takahara, Minimum cost guaranteed lifetime design for heterogeneous wireless sensor networks, in: Proceedings of IEEE International Workshop on Strategies for Energy Efficiency in Ad-hoc and Sensor Networks, IPCCC, 2005.

[19] Q. Wang, K. Xu, G. Takahara, H. Hassanein, Locally optimal relay node placement in heterogeneous wireless sensor networks, in: Proceedings of IEEE Globecom, 2005.

[20] K. Xu, Q. Wang, H. Hassanein, G. Takahara, Optimal design of wireless sensor networks: minimum cost with lifetime constraints, in: Proceedings of IEEE WiMob, 2005.

[21] O. Younis, S. Fahmy, Distributed clustering for scalable, long-lived sensor networks, Technical Report TR-03-026, Purdue University, 2003.

Sylvia Tai obtained her M.Sc. degree in the School of Computing at Queen's University. She graduated with a B.Math (Honours, Co-op) in Computer Science from the University of Waterloo in 2003. Her research interests include wireless sensor networks, ad hoc networks, and issues in network planning and resource management.

Robert Benkoczi received his Ph.D. in Computer Science from Simon Fraser University in 2004. He currently is a Postdoctoral Fellow in the School of Computing at Queen's University. His research is in combinatorial optimization with focus on facility location and the management of wireless networks.

Hossam Hassanein is a Professor in the School of Computing at Queen's University in the areas of broadband, wireless and variable topology networks architecture, protocols, control and performance evaluation. He obtained his Ph.D. in Computing Science from the University of Alberta in 1990. He is the founder and director of the Telecommunication Research (TR) Lab (http://www.cs.queensu.ca/trl) in the School of Computing at Queen's. Dr. Hassanein has more than 200 publications in reputable journals, conferences and workshops. He is a senior member of the IEEE and serves on the program committee of a numerous international conferences and workshops.

Selim G. Akl received his Ph.D. degree from McGill University in Montreal in 1978. He is currently a Professor of Computing at Queen's University, Kingston, Ont., Canada. His research interests are in parallel computation. He is the author of *Parallel Sorting Algorithms* (Academic Press, 1985), *The Design and Analysis of Parallel Algorithms* (Prentice-Hall, 1989), and *Parallel Computation*: *Models and Methods* (Prentice-Hall, 1997), and a co-author of *Parallel Computational Geometry* (Prentice-Hall, 1992). Dr. Akl presently serves on the editorial boards of *Computational Geometry*, *Communications in Applied Geometry*, *Parallel Processing Letters*, and the *International Journal of Parallel*, *Emergent*, *and Distributed Systems*, and is a founding editorial board member of the *International Journal of High Performance Computing and Networking*.