# Maximizing Message Routing Requests in Wireless Sensor Networks

Zane Sumpter[1], Lucas Burson[1], Xiao Chen[2], Bin Tang[3],
[1]Department of Electrical Engineering and Computer Science,
Wichita State University, Wichita, KS, USA
[2]Department of Computer Science, Texas State University, San Marcos, TX, USA
[3]Department of Computer Science, Azusa Pacific University, Azusa, CA, USA
Email: zpsumpter, ljburson@wichita.edu, xc10@txstate.edu, btang@apu.edu

***Abstract*— In this paper, we study energy-efficient routing problem in wireless sensor networks with the objective of maximizing the number of satisfiable requests under the constraint that sensors have finite battery power. We refer to the problem as $maxR$. The online version of the problem, where a sequence of messages that has to be routed over the network is not known ahead of time, has been studied by several researchers. In this paper, we propose a new online algorithm called SIMPLE to improve the results with simple computations. We also study the offline version of $maxR$ where the sequence of requests is pre-known. As far as we know, the offline $maxR$ problem has not been well studied. After appropriate transformation, the offline $maxR$ problem is equivalent to the well-known maximum disjoint path problem, which is NP-hard. We put forward a greedy offline algorithm called GDP and use it as a benchmark to compare with the online algorithms. Simulation results show that the SIMPLE algorithm outperforms the OML algorithm in the number of satisfiable requests and is very close to the benchmark. SIMPLE and GDP are also better than OML in terms of the number of energy-depleted nodes and energy balancing among sensor nodes, and perform comparably with OML in terms of total energy consumption.**

**Keywords** – Energy efficiency, greedy algorithm, heuristics, message routing, wireless sensor networks

## I. INTRODUCTION

A wireless sensor network (WSN) is a computer network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. Sensor networks are used in military applications as well as many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control. These applications require great care in the utilization of power. The power level is provided by batteries of sensors and thus it is finite. Every message sent and every computation performed drains the battery.

Because of the limited power in sensors and the inconvenience to recharge their batteries frequently, energy efficiency has always been a consideration for routing algorithms in sensor networks. Routing is a process to send a message from a source to a destination. After the sensors are deployed and form a network, pairs of sensors exchange messages in an unpredictable sequence. For proper network operation, it is critical that every attempt to transmit a message succeed. Hence, we are interested in designing energy-efficient routing algorithms with the objective of maximizing the number of messages (requests) successfully routed before the first failed message route. We refer to the problem as $maxR$ in this paper. The maximum number of messages that a network can deliver before first failure reflects the *lifetime* of a sensor network and several researchers have proposed heuristics ($zP_{min}$ [1], CMAX [7], MRPC [11], OML [12], MMBCR [15]) to maximize it. All of these are *online* algorithms meaning they *do not know ahead of time the sequence of messages that has to be routed over the network*. Among them, the OML algorithm proposed by Park and Sahni performs the best in terms of maximizing the number messages delivered, energy consumption and energy balancing.

However, we find that the OML algorithm is complex and whose parameters are difficult to set in real applications (see Section IV), so we propose a simpler online heuristic routing algorithm called *SIMPLE* to maximize the number of satisfiable requests in this paper. Also, we address the *offline* version of the $maxR$ problem where we maximize the number of satisfiable requests by assuming that the sequence of requests is pre-known. In that case, we can route as many messages as possible regardless of the requests order. To the best of our knowledge, this problem has not been well studied. We find that after appropriate transformation, the offline $maxR$ problem is equivalent to the well-known maximum disjoint path problem [8], which is NP-hard. We propose an offline *Greedy-Disjoint-Paths* (GDP) algorithm to address $maxR$. *It is known that the best offline algorithm performs better than any online algorithm*. Therefore, we can use GDP as a benchmark to compare with the online algorithms. Simulation results show that the SIMPLE algorithm outperforms the OML algorithm in the number of satisfiable requests and is very close to the benchmark. SIMPLE and GDP are also better than OML in terms of the number of energy-depleted nodes and energy balancing among sensor nodes, and perform comparably with OML in terms of total energy consumption.

The rest of the paper is organized as follows: Section II introduces the preliminary. Section III formulates the $maxR$

| Notation | Description |
| --- | --- |
| $ie(u) > 0$ | initial energy in sensor $u$ |
| $ce(u) \geq 0$ | the current energy in sensor $u$ |
| $w(u,v) > 0$ | the energy required for a single-hop transmission from sensor $u$ to $v$ |
| $re(u)$ | the remaining energy in sensor $u$ |

TABLE I
THE NOTATIONS

problem. Section IV references the related works. Section V presents our proposed algorithms. Section VI compares all algorithms by simulation. And Section VII concludes the paper and points out the future work.

## II. PRELIMINARY

### A. Network and Energy Models

A sensor network can be represented by a general undirected graph $G(V, E)$, where $V = \{1, 2, ..., N\}$ is a set of sensor nodes and $E$ is a set of edges. There is an edge $(u, v) \in E$ connecting sensor $u$ and sensor $v$ iff a single-hop transmission between $u$ and $v$ is possible.

We assume that each sensor node $i$ has a finite and unreplenishable initial energy $ie(i)$, which is a non-negative integer value. For the energy consumption of sending and receiving a message by a node, we adopt the first order radio model [4] where for $k$-bit data over distance $l$, the transmission energy $E_T(k, l) = E_{elec} \times k + \epsilon_{amp} \times k \times l^2$, and the receiving energy $E_R(k) = E_{elec} \times k$, where $E_{elec} = 50nJ/bit$ and $\epsilon_{amp} = 100pJ/bit/m^2$. When the distances among senors are in the order of one hundred meters, the term with $\epsilon_{amp}$ is much larger than the term with $E_{elec}$. Therefore, we assume that for each node, sending one unit-sized message costs one unit of energy while receiving one message costs zero energy. This assumption is adopted also for the purpose of fair comparison later - Park and Sahni [12] also assume no energy consumption during message reception in OML.

### B. Notations

The notations used in the paper are summarized in Table I.

## III. PROBLEM FORMULATION

We formulate the $maxR$ problem as follows: There are a set of $p$ routing requests $\mathcal{R} = \{r_1, r_2, \cdots, r_p\}$ in the network where each request $r_i = (s_i, t_i)$ represents a message sending from source $s_i$ to destination $t_i$, $1 \leq i \leq p$. We assume that each message is of unit size. The objective of the problem is to complete the maximum number of routing requests under the energy constraint that $re(u) \geq 0$, $\forall u \in V$. This constraint implies that any node can not spend more energy than its initial energy level.

The online version of $maxR$ assumes that the sequence of messages that has to be routed over the network is not known ahead of time while the offline version assumes the sequence is pre-known. The major difference in algorithm design between the two is that the online algorithms have to keep the order of the requests while the offline ones do not have to.

## IV. RELATED WORKS

The online $maxR$ problem is one approach to realize energy-efficient routing in sensor networks. Several authors have developed energy-efficient algorithms [2], [5], [6], [10], [13]–[16] either through the lifetime (time at which a communication fails first) or through capacity (the number of successful communications over some fixed period of time).

Toh et al. [15] propose the MMBCR (min-max battery cost routing) algorithm to select a source-to-destination path. The MMBCR algorithm selects a path for which the minimum of the residual energies of the sensors on the path is maximum. They also propose a conditional MMBCR algorithm CMMBCR to balance between the energy consumed by a route and the minimal residual energy at the nodes along the chosen route. Misra and Banerjee [11] put forward the MRPC (maximum residual packet capacity) lifetime-maximization heuristic where routing is done along a path with maximum lifetime. A conditional MRPC algorithm CMRPC is also proposed and attempts to balance energy consumption. Aslam et al. [1] propose the max-min $zP_{min}$-path algorithm to select routes that attempt to balance energy. This algorithm selects a path that uses at most $z * P_{min}$ energy, where $z$ is a parameter of the algorithm and $P_{min}$ is the energy required by the minimum-energy path. The selected path maximizes the minimum residual energy fraction (energy remaining after route/initial energy) for nodes on the route path. Kar et al. [7] develop a capacity-competitive (the capacity is the number of messages routed over some time period) algorithm, CMAX (capacity maximization), with logarithmic competitive ratio. To achieve logarithmic competitive ratio, the CMAX algorithm does admission control, that is, it rejects some routes that are possible. The CMAX algorithm has a complexity advantage over the max-min $zP_{min}$ algorithm and experimental results in [7] suggest that CMAX with no admission control outperforms max-min $zP_{min}$ on both the lifetime and capacity metrics.

To the best of our knowledge, the algorithm that outperforms the above algorithms in terms of lifetime and energy consumption is the OML algorithm put forward by Park and Sahni [12] (see Fig. 1). This is the algorithm that we will compare our algorithms with. To make the paper self-inclusive, we introduce the OML algorithm in detail as follows.

The main idea of OML is that in order to maximize lifetime, delay as much as possible the depletion of a sensor's energy. OML achieves this by a two-step process to find a path for each routing request $r_i = (s_i, t_i)$. In the first step, $G$ is transformed to $G'$ by removing all edges $(u, v)$ which require more energy than available for a transmit. Next, determine the minimum energy path, $P'_i$ from $s_i$ to $t_i$ in the pruned graph $G'$. This may be done using Dijkstra's shortest path algorithm [3]. If no such path exists, the routing request $r_i$ fails. Otherwise, using $P'_i$, compute the residual energy, $re(u) = ce(u) - w(u, v)$ for $(u, v)$, an edge on $P'_i$. Let $minRE = min\{re(u) | u \in P'_i$ and $u \neq t_i\}$. Let

**Algorithm OML**: Online Maximum Lifetime heuristic algorithm

**for** each request $r_i \in \mathcal{R} = \{r_1, r_2, \cdots, r_p\}$ **do**
  **Step 1:** [Compute $G''$]
  $G' = (V, E')$ where $E' = E - \{(u, v)|ce(u) < w(u, v)\}$.
  Let $P_i'$ be a shortest $s_i$ to $t_i$ path in $G'$.
  If there is no such $P_i'$, the route request fails, stop.
  Compute the minimum residual energy $minRE$ for sensors other than $t_i$ on $P_i'$.
  Let $G'' = (V, E'')$ where $E'' = E' - \{(u, v)|ce(u) - w(u, v) < minRE\}$.
  **Step 2**: [Find route path]
  Compute the weight $w''(u, v)$ for each edge of $E''$.
  Let $P_i''$ be a shortest $s_i$ to $t_i$ path in $G''$.
  Use $P_i''$ to route from $s_i$ to $t_i$.
**end for**

Fig. 1. The OML algorithm

$G'' = (V, E'')$ be obtained from $G'$ by removing all edges $(u, v) \in E'$ with $ce(u) - w(u, v) < minRE$. That is, all edges whose use would result in a residual energy below $minRE$ are pruned from $E'$. This pruning is an attempt to prevent the depletion of energy from sensors that are low on energy.

In the second step, find the path to route request $r_i$. For this, we begin with $G''$ as above and assign weights to each $(u, v) \in E''$. The weight assignment is done so as to balance the desire to minimize total energy consumption as well as the desire to prevent the depletion of a sensor's energy. Let $eMin(u) = min\{w(u, v)|(u, v) \in E''\}$ be the energy needed by sensor $u$ to transmit a message to its nearest neighbor in $G''$. Let $\rho$ be defined as below.

$$\rho(u, v) = \begin{cases} 0 & \text{if } ce(u) - w(u, v) > eMin(u) \\ c & \text{otherwise,} \end{cases}$$

where $c$ is a nonnegative constant and is an algorithm parameter. For each $u \in V$, define

$$\alpha(u) = \frac{minRE}{ce(u)}.$$

The weight $w''(u, v)$ assigned to edge $(u, v) \in E''$ is

$$w''(u, v) = (w(u, v) + \rho(u, v))(\lambda^{\alpha(u)} - 1),$$

where $\lambda$ is another nonnegative constant and an algorithm parameter. As can be seen, this weighting function, through $\rho$, assigns a high weight to edges whose use on a routing path cause a sensor's residual energy to become low. Also, all edges emanating from a sensor whose current energy is small relative to $minRE$ are assigned a high weight because of the $\lambda$ term. Thus, the weighting function discourages the use of edges whose use on a routing path is likely to result in the failure of a future route.

Through the study of OML, we find that OML is complex and whose parameters such as $c$ and $\lambda$ are difficult to set

**Algorithm SIMPLE**: a simple heuristic method to maximize the number of completed requests

**for** each request $r_i \in \mathcal{R} = \{r_1, r_2, \cdots, r_p\}$ **do**
  $G' = (V, E')$ where $E' = E - \{(u, v)|ce(u) < w(u, v)\}$.
  In $G'$, find the path $\{s_i, u_1, u_2, \cdots, u_q, t_i\}$ from $s_i$ to $t_i$ that can minimize metric $\sum_{j=1}^{q} \frac{1}{ce(u_j)}$;
  **if** the returned path is NULL **then**
    Stop the program
  **end if**
  **for** each node $u_i$ on the path found except the sink node **do**
    $ce(u_i) = ce(u_i) - w(u_i, u_{i+1})$
  **end for**
**end for**

Fig. 2. The SIMPLE algorithm

in real applications. That motivates us to put forward a new online heuristic algorithm to improve the performance with simpler computation. Furthermore, we are interesting in addressing the unexplored offline $maxR$ problem and using the offline algorithm as a benchmark to compare with the online algorithms.

## V. The Algorithms

In this section, we first put forward a new online heuristic algorithm SIMPLE with simpler computation and then propose an offline greedy algorithm GDP to address the $maxR$ problem.

### A. The SIMPLE Algorithm

The SIMPLE algorithm is shown in detail in Fig. 2. It tries to satisfy the maximum number of requests through minimizing total energy consumption and energy balancing in message routing: on one side, it minimizes the total energy consumption by finding the shorter path between source and sink nodes and on the other side, it favors nodes with higher remaining energy. The SIMPLE algorithm achieves these goals by finding a path between the source and destination that can minimize the metric $\sum_{j=1}^{q} \frac{1}{ce(u_j)}$. Intuitively, a shorter path with higher energy nodes will be selected to route the message. As OML, SIMPLE first deletes those edges that require more energy than available for a transmit. Then, choosing a path that minimizes the metric between a source and a destination is simple. For example, suppose there are two paths between $s_i$ and $t_i$: $\{s_i, u_1, u_2, t_i\}$ and $\{s_i, u_3, u_4, t_i\}$. The current energy levels of $u_1, u_2, u_3, u_4$ are $2, 3, 3, 4$, respectively. According to the metric formula, the metric for the first path is: $\frac{1}{2} + \frac{1}{3} = \frac{5}{6}$ and the metric for the second path is: $\frac{1}{3} + \frac{1}{4} = \frac{7}{12}$. The second metric is smaller, so the second path will be chosen.

### B. The Greedy Disjoint Path (GDP) Algorithm

Before presenting the algorithm, we first transform an undirected graph $G(V, E)$ into a new directed graph $G'(V', E')$,

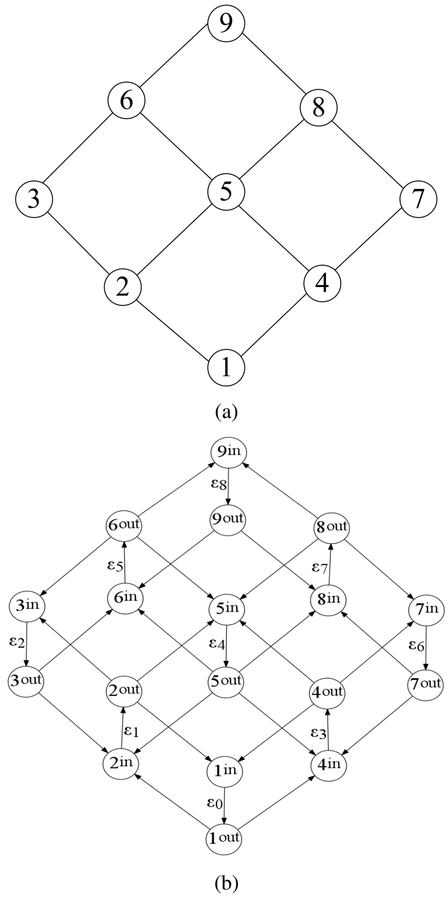



Fig. 3. A sensor network before and after transformation. There are nine nodes and three source-destination pairs: $(3, 8)$, $(6, 4)$, and $(7, 2)$ before transformation (a); or $(3^{in}, 8^{in})$, $(6^{in}, 4^{in})$, and $(7^{in}, 2^{in})$ after transformation (b). The capacity of edge $(i^{in}, i^{out})$ is $\epsilon_i$, the initial energy of node $i$. The capacity of other edges in the transformed graph is infinity.

and show that the $maxR$ on $G'$ is equivalent to the well-known maximum disjoint path problem [8].

*1) Graph Transformation:* First, replace each undirected edge $(i, j) \in E$ with two directed edges $(i, j)$ and $(j, i)$, and set the capacity of all the directed edges as infinity. Then split each node $i \in V$ into two nodes: in-node $i^{in}$ and out-node $i^{out}$, and add a directed edge $(i^{in}, i^{out})$ with capacity as $\epsilon_i$, the initial energy of node $i$. All the incoming directed edges of node $i$ are incident on $i^{in}$ and all the outgoing directed edges of node $i$ emanate from $i^{out}$. Now the initial routing requests $(s_1, t_1), (s_2, t_2), ..., (s_p, t_p)$ in $G$ become new routing requests $(s_1^{in}, t_1^{in}), (s_2^{in}, t_2^{in}), ..., (s_p^{in}, t_p^{in})$ in $G'$. Note that it is the in-node of each destination node that becomes the new destination node, due to receiving messages not costing energy in our model. Assume that there are $m'$ and $n'$ number of edges and nodes in the transformed graph, then $n' = 2n$ and $m' = n + 2m$.

We use Fig 3 to give an example of a sensor network before and after the transformation. It is a $3 \times 3$ grid network, with three source-destination pairs: $(3, 8)$, $(6, 4)$, and $(7, 2)$. After transformation, the three source-destination pairs are: ($3^{in}$, $8^{in}$), ($6^{in}$, $4^{in}$), and ($7^{in}$, $2^{in}$).

**Algorithm GDP**: a Greedy-Disjoint-Paths algorithm to find the maximum number of completed requests on $G'(V', E')$

**Notations**: $m'$ is the total number of edges in $G'$; $\epsilon$ is the initial energy of all the nodes; $\beta = m'^{1/\epsilon+1}$
$\mathcal{I} = \emptyset$, $\mathcal{I}$ is the set of completed requests
For all $e \in E'$, set its weight to 1
**while** There are still requests that can be satisfied **do**
  Let $P_i$ be the minimum weighted path so that adding $P_i$ to the selected set of paths does not use any edge more than $\epsilon$ times, and $P_i$ connects some $(s_i, t_i)$ pair not yet connected
  Add $i$ to $\mathcal{I}$ and use path $P_i$ to route the message from $s_i$ to $t_i$
  Multiply the length of all edges along $P_i$ by $\beta$
**end while**

Fig. 4. The GDP algorithm

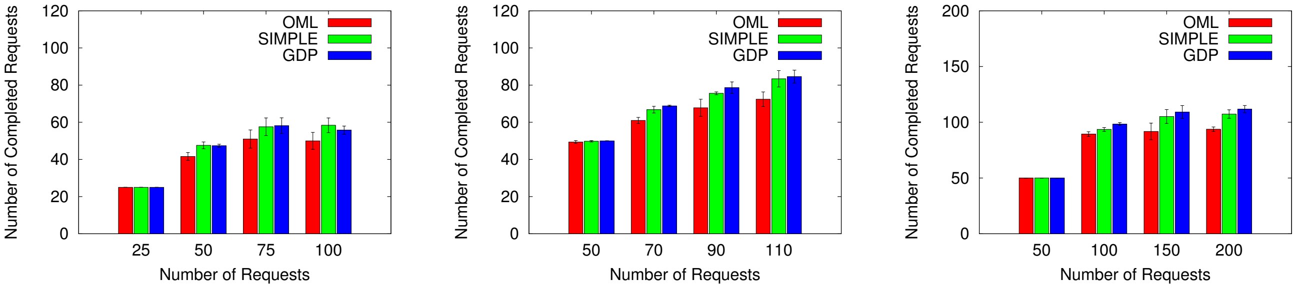
The theorem below states that if all the nodes have the same initial energy level $\epsilon$, then $maxR$ becomes the well-known maximum disjoint path problem [8], [9].

*Theorem 1:* If the initial energy levels of all the sensor nodes are equaled to $\epsilon$, finding the maximum number of satisfiable routing requests in $G'(V', E')$ is equivalent to the maximum disjoint path problem in $G'(V', E')$.

*Proof:* The maximum disjoint path problem is as follows. Given a directed graph $G'(V', E')$ and an integer capacity $c$ of each edge, and let $\mathcal{R}$ be the set of $p$ connection request pairs $\{(s_1, t_1), ..., (s_p, t_p)\}$. A subset $\mathcal{I}$ of $\mathcal{R}$ is realizable in $G'$ if all the requests in $\mathcal{I}$ can be satisfied while each edge in $G'$ is used by at most $c$ times. The goal of the maximum disjoint path problem is to find the maximum size of a realizable subset $\mathcal{I}$ of $\mathcal{R}$.

In $maxR$, assume that the initial energy levels of all the sensor nodes are equaled to $\epsilon$. Then finding the maximum number of satisfiable routing requests from $(s_1^{in}, t_1^{in}), (s_2^{in}, t_2^{in}), ..., (s_p^{in}, t_p^{in})$ in $G'(V', E')$ is exactly the maximum disjoint path problem in $G'(V', E')$ with $\mathcal{R} = \{(s_1^{in}, t_1^{in}), (s_2^{in}, t_2^{in}), ..., (s_p^{in}, t_p^{in})\}$ and $\epsilon = c$. □

The maximum disjoint path problem is NP-hard [8], [9]. Following the idea of Theorem 1 and inspired by the works in [8], [9], we present a greedy algorithm called Greedy-Disjoint-Paths (GDP) algorithm in Fig. 4. In GDP, it always tries to find a minimum weighted path connecting a source and destination node pair while still satisfying the capacity of each edge. Once that minimum weighted path is selected, the weights of all the edges on the path are multiplied by $\beta$ ($\beta > 1$). Here, $\beta = m'^{1/\epsilon+1}$, where $m'$ is the total number of edges in $G'$ and $\epsilon$ is the initial energy of all the nodes. The intuition behind setting $\beta$ like this is that an edge will gain more weight if it has been used already, which encourages other edges to be used to route following messages.

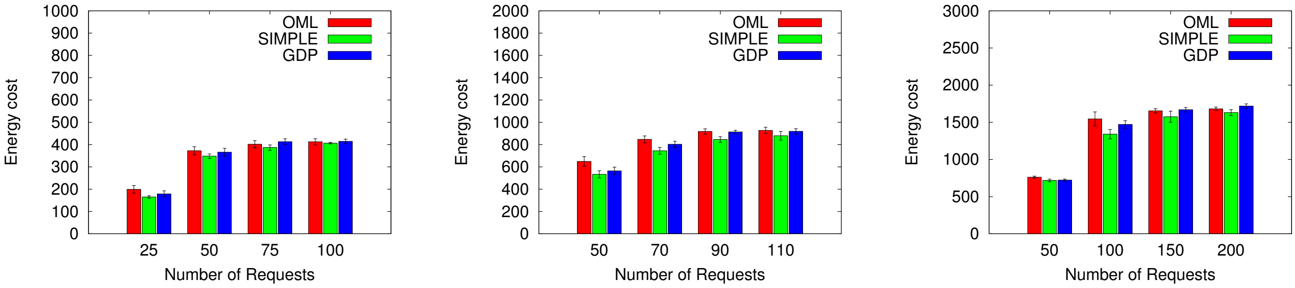



(a) 10x10 Network.

(b) 15x15 Network.

(c) 20x20 Network.

Fig. 5. Number of satisfiable requests in the network.




(a) 10x10 Network.

(b) 15x15 Network.

(c) 20x20 Network.

Fig. 6. Total energy consumption of sensor nodes.

## VI. Simulation

We compare our algorithms SIMPLE and GDP with OML using a self-written simulator in C language. We assume that sensor nodes are connected in a grid. The reason to use a grid network is because it simplifies algorithm implementation without compromising the results. All our algorithms can be applied to a general network topology. In a grid, we assume the distance between any two neighbors is one unit.

### A. Parameter setting

In our simulations, we use three settings of grid network topology: $10 \times 10$, $15 \times 15$ and $20 \times 20$, with $100$, $225$, and $400$ sensor nodes, respectively. We randomly generate a sequence of requests $\mathcal{R} = \{r_1, r_2, \cdots, r_p\}$ with sources and destinations randomly selected. In all simulation figures below, each data point is an average of five runs and the error bars indicate $95\%$ confidence interval. In comparing all the three algorithms: OML, SIMPLE, and GDP, we set the initial energy of all nodes to $5$ so that the request sequence cannot be finished by all the algorithms.

In setting the parameters for OML, since we use a grid network topology, without loss of generality, we set the energy consumption $w(u, v)$ to send a message from $u$ to $v$ to $1$. The $eMin(u)$ in OML, which is the energy needed by $u$ to transmit a message to its nearest neighbor, also equals $1$. Then $\rho(u, v)$ in OML is $0$. The algorithm parameter $\lambda$ is set to $10^{11}$ because it gives a stable performance of OML [12].

### B. Number of completed requests

Figure 5 shows the number of requests completed (satisfied) for all three algorithms in $10 \times 10$, $15 \times 15$, and $20 \times 20$ grid networks, respectively. The results show that when the number of requests is small (e.g. 25), all algorithms can complete all the requests, indicating that the performance difference of the algorithms is small in a less stressful scenario. However, when the number of requests increases, the GDP algorithm satisfies the most number of message requests. SIMPLE outperforms OML and whose results are very close to those of GDP.

### C. Total energy consumption

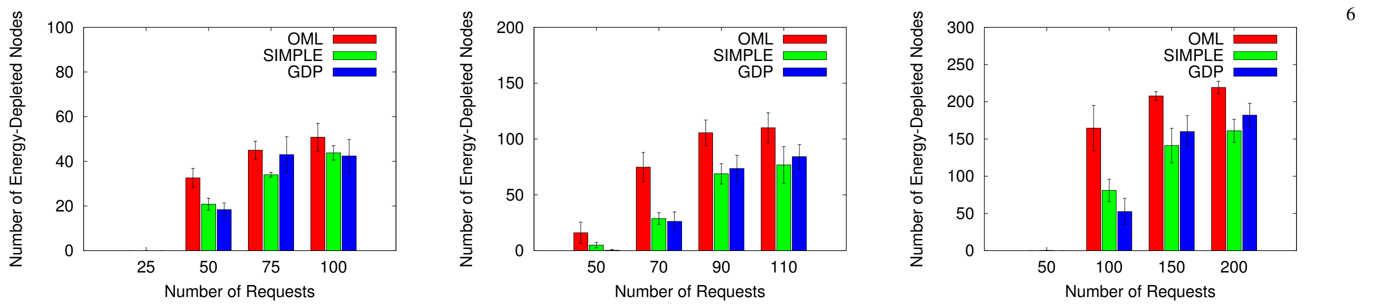
The comparison of the total energy consumption is shown in Figure 6. The results show similar performance of all three algorithms, with GDP and SIMPLE consuming slightly less total energy than OML.

### D. Number of energy-depleted nodes

Figure 7 shows the number of energy-depleted nodes resulted from the three algorithms. It shows that in all the cases, the number of energy-depleted nodes in OML is much larger than that in the other two algorithms. This demonstrates that both SIMPLE and GDP algorithms perform better energy balancing among sensor nodes than OML.

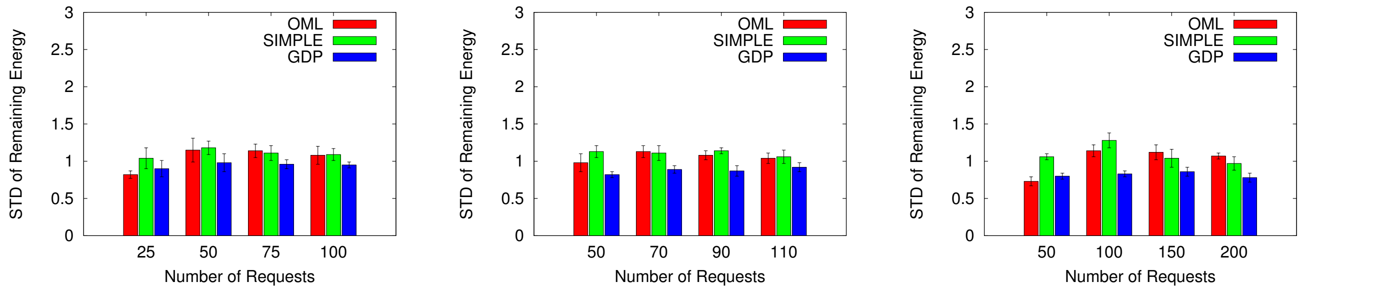
### E. Standard deviation of remaining energy

To further investigate how balanced the remaining energy of the nodes is, we calculate the standard deviation (STD) of the remaining energy of all the nodes. Figure 8 shows that




(a) 10x10 Network. (b) 15x15 Network. (c) 20x20 Network.

Fig. 7. Number of energy-depleted nodes.




(a) 10x10 Network. (b) 15x15 Network. (c) 20x20 Network.

Fig. 8. Standard deviation of remaining energy of sensor nodes.

GDP balances the energy levels in nodes the best in routing, while OML and SIMPLE perform similarly.

## VII. Conclusion and Future Direction

In this paper, we developed energy-efficient routing algorithms with the objective of maximizing the number of satisfiable requests under the constraint of limited battery power of sensor nodes. We first proposed a new online algorithm SIMPLE and then put forward an offline greedy algorithm GDP as a benchmark. We showed empirically that SIMPLE outperforms the OML algorithm in the number of satisfiable requests and is very close to the benchmark. SIMPLE and GDP are also better than OML in terms of the number of energy-depleted nodes and energy balancing among sensor nodes, and perform comparably with OML in terms of total energy consumption. In the future, we will design the distributed versions of our proposed algorithms and explore new energy-efficient routing algorithms in sensor networks.

## References


[1] J. Aslam, Q. Li, and D. Rus, "Three Power-Aware Routing Algorithms for Sensor Network," *Wireless Comm. and Mobile Computing*, 3, 2003, pp. 187-208.

[2] J. Chang and L. Tassiulas, "Fast Approximate Algorithms for Maximum Lifetime Routing in Wireless Ad-Hoc Networks," *Proc. IFIP-TC6 Networking*, 2000, pp. 702-713.

[3] M. L. Fredman and R. E. Tarjan, "Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms," *Proc. of the 25th IEEE Annual Symp. on Foundations of Computer Science*, 1984, pp. 338-346.

[4] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient Communication Protocol for Wireless Microsensor Networks," *Proc. of HICSS*, 2000.

[5] R. Kannan, L. Ray, R. Kalidindi, and S. Iyengar, "Energy Threshold Constrained Geographic Routing Protocol for Wireless Sensor Networks," *Signal Processing Letters*, vol. 1, no. 1, 2003, pp. 79-84.

[6] R. Kannan and S. Iyengar, "Game Theoretic Models for Reliable Path-Length and Energy Constrained Routing with Data Aggregation in Wireless Sensor Networks," *IEEE J. Selected Areas in Comm.*, vol. 22, no. 6, 2004, pp. 1141-1150.

[7] K. Kar, M. Kodialam, T. Lakshman, and L. Tassiulas, "Routing for Network Capacity Maximization in Energy-Constrained Ad-Hoc Networks," *Proc. of IEEE INFOCOM*, 2003.

[8] J. Kleinberg, "Approximation Algorithms for Disjoint Paths Problems," *PH.D. Dissertation*, 1996.

[9] J. Kleinberg and E. Tardos, "Algorithm Design," Addison Wesley, 2005.

[10] M. Maleki, K. Dantu, and M. Pedram, "Power-aware Source Routing Protocol for Mobile Ad Hoc Networks," *Proc. International Symposium Low Power Electronics and Design (ISLPED)*, 2002, pp. 72-75.

[11] A. Misra and S. Banerjee, "MRPC: Maximizing Network Lifetime for Reliable Routing in Wireless Environments," Proc. of IEEE Wireless Communication and Networking Conference (WCNC), 2002.

[12] J. Park and S. Sahni, "An Online Heuristic for Maximum Lifetime Routing in Wireless Sensor Networks," *IEEE Trans. on Computers*, vol. 55, no. 8, Aug. 2006, pp. 1048-1052.

[13] A. Spyropoulos and C. Raghavendra, "Energy Efficient Communications in Ad Hoc Networks Using Directional Antenna," *Proc. IEEE INFOCOM*, 2002.

[14] I. Stojmenovic and X. Lin, "Power-Aware Localized Routing in Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 12, no. 11, Nov. 2001, pp. 1122-1133.

[15] C. Toh, H. Cobb, and D. Scott, "Performance Evaluation of Battery-Life-Aware Routing Optimization Schemes for Wireless Ad Hoc Networks," *Proc. IEEE International Computing Conference. (ICC)*, 2001.

[16] G. Zussman and A. Segall, "Energy Efficient Routing in Ad Hoc Disaster Recovery Networks," *Proc. IEEE INFOCOM*, 2003.