5

WSN Design for Unlimited Lifetime

Emanuele Lattanzi and Alessandro Bogliolo
DiSBeF - University of Urbino
Italy

1. Introduction

Wireless sensor networks (WSNs) are among the most natural applications of energy harvesting techniques. Sensor nodes, in fact, are usually deployed in harsh environments with no infrastructured power supply and they are often scattered over wide areas where human intervention is difficult and expensive, if not impossible at all. As a consequence, their actual lifetime is limited by the duration of their batteries, so that most of the research efforts in the field of WSNs have been devoted so far to lifetime maximization by means of the joint application of low-power design, dynamic power management, and energy-aware routing algorithms. The capability of harvesting renewable power from the environment provides the opportunity of granting unbounded lifetime to sensor nodes, thus overcoming the limitations of battery-operated WSNs. In order to optimally exploit the potential of energy-harvesting WSNs (hereafter denoted by EH-WSNs) a paradigm shift is required from energy-constrained lifetime maximization (typical of battery-operated systems) to power-constrained workload maximization. As long as the average workload at each node can be sustained by the average power it takes from the environment (and environmental power variations are suitably filtered-out by its onboard energy buffer) the node can keep working for an unlimited amount of time. Hence, the main design goal for an EH-WSN becomes the maximization of its sustainable workload, which is strongly affected by the routing algorithm adopted.

It has been shown that EH-WSNs can be modeled as generalized flow networks subject to capacity constraints, which provide a convenient representation of power, bandwidth, and resource limitations (Lattanzi et al., 2007). The maximum sustainable workload (MSW) for a WSN is the so called *maxflow* of the corresponding flow network. Four main results have been recently achieved under this framework (Bogliolo et al., 2010). First, given an EH-WSN and the environmental conditions in which it operates, the theoretical value of the maximum energetically sustainable workload (MESW) can be exactly determined. Second, MESW can be used as a design metric to optimize the deployment of EH-WSNs. Third, the energy efficiency of existing routing algorithms can be evaluated by comparing the actual workload they can sustain with the theoretical value of MESW for the same network. Fourth, self-adapting maxflow (SAMF) routing algorithms have been developed which are able to route the MESW while adapting to time-varying environmental conditions.

This chapter introduces the research field of *design for unlimited lifetime* of EH-WSNs, which aims at exploiting environmental power to maximize the workload of the network under steady-state sustainability constraints. The power harvested at each node is regarded as a

time-varying constraint of an optimization problem which is defined and addressed within the theoretical framework of generalized flow networks. The solution of the constrained optimization problem provides the best strategy for managing the network in order to obtain maximum outputs without running out of energy.

The following subsection provides a brief overview of previous work on routing algorithms for autonomous WSNs. Section 2 presents the network model used for analyzing workload sustainability under energy, bandwidth and resource constraints, and introduces the concept of maximum sustainable workload; Section 3 outlines the SAMF routing algorithm, demonstrating its optimality and highlighting its theoretical properties, Section 4 introduces design and simulation tools based on workload sustainability, while Section 5 discusses the practical applicability of SAMF algorithms in light of simulation results obtained by taking into account the effects of non-idealities such as finite propagation time, radio broadcasting, radio channel contention, and packet loss.

1.1 Previous work

The wide range of possible applications and operating environments of *wireless sensor networks* (WSNs) makes scalability and adaptation essential design goals (Dressler, 2008) which have to be achieved while meeting tight constraints usually imposed to sensor nodes in terms of size, cost, and lifetime (Yick et al., 2008). Since the main task of any WSN is to gather data from the environment, the routing algorithm applied to the network is one of the most critical design choices, which has a sizeable impact on power consumption, performance, dependability, scalability, and adaptation. The operation of any WSN usually follows a 2-phase paradigm. In the first phase, called *dissemination*, control information is diffused in order to dynamically change the sampling task (which can be specified in terms of sampling area, target nodes, sampling rate, sensed quantities, ...); in the second phase, called *collection*, sampled data are transmitted from the involved sensor nodes to one or more collection points, called *sinks* (Levis et al., 2008). Routing algorithms have e deep impact on both dissemination and collection phases.

Energy efficiency is a primary concern in the design of routing algorithms for WSNs (Mhatre & Rosenberg, 2005; Shafiullah et al., 2008; Yarvis & Zorzi, 2008). If the routing algorithm requires too many control packets, chooses sub-optimal routes, or requires too many computation at the nodes, it might end up reducing the lifetime of the network because of the limited energy budget of battery-operated sensor nodes. The routing algorithms which have been proposed to maximize network lifetime are documented in many comprehensive surveys (Chen & Yang, 2007; Li et al., 2011; Yick et al., 2008).

Taking a different perspective, lifetime issues can be addressed by means of energy harvesting techniques, which enable the design of autonomous sensor nodes taking their power supply from renewable environmental sources such as sun, light, and wind (Amirtharajah et al., 2005; Nallusamy & Duraiswamy, 2011; Sudevalayam & Kulkarni, 2010). Environmentally-powered systems, however, give rise to additional design challenges due to supply power uncertainty and variability.

While there are a number of routing protocols designed for battery-operated WSNs, only a small number of routing protocols have been published which explicitly account for energy

harvesting. *Geographic routing* algorithms (Eu et al., 2010; Zeng et al., 2006) take into account distance information, link qualities, and environmental power at each node in order to select the best candidate region to relay a data packet. Both algorithms, however, strongly depend on the position awareness of sensor nodes, which is difficult to achieve in many WSNs. The environmental power available at each node is used as a weight in the *energy-opportunistic weighted minimum energy* (E-WME) algorithm to determine the weighted minimum path to the sink (Lin et al., 2007).

Moving beyond the opportunity of exploiting environmental power to recharge energy buffers and enhance lifetime, energy harvesting prompt for a paradigm shift from *energy-constrained lifetime maximization* to *power-constrained workload optimization*. In fact, as long as the average workload at each node can be sustained by the average power it takes from the environment, the node can keep working for an unlimited amount of time. In this case rechargeable batteries are still used as energy buffers to compensate for environmental power variations, but their capacity does not affect any longer the lifetime of the network.

It has been shown that autonomous wireless sensor networks can be modeled as flow networks (Bogliolo et al., 2006), and that the *maximum energetically sustainable workload* (MESW) can be determined by solving an instance of *maxflow* (Ford & Fulkerson, 1962). The solution of maxflow induces a MESW-optimal *randomized minimum path recovery time* (R-MPRT) routing algorithm that can be actually implemented to maximally exploit the available power (Lattanzi et al., 2007). Different versions of the R-MPRT algorithm have been proposed to improve performance and reduce packet loss in real-world scenarios, taking into account MAC protocol overhead and lossy wireless channels (Hasenfratz et al., 2010). Environmental changes, however, impose to periodically recompute the global optimum and to update the routing tables of R-MPRT algorithms. A distributed version of maxflow has been proposed that exploits the computational power of WSNs (Kulkarni et al., 2011) to grant to the network the capability of recomputing its own routing tables for adapting to environmental changes (Klopfenstein et al., 2007). Adaptation, however, is a complex task which might conflict with the normal operations of the WSN, thus imposing to trade off adaptation frequency for availability. In general, the adaptation and scalability needs which are typical of WSNs prompt for the application of some sort of self-organization mechanisms (Dressler, 2008; Eu et al., 2010; Mottola & Picco, 2011). In particular, a self-adapting maxflow routing strategy for EH-WSNs has been recently proposed (Bogliolo et al., 2010) which is able to route the maximum sustainable workload under time-varying power, bandwidth, and resource constraints.

2. Network model and workload sustainability

Any WSN can be modeled as a directed graph with vertices associated with network nodes and edges associated with direct links among them: vertices v_i and v_j are connected by an edge $e_{i,j}$ if and only if there is a wireless connection from node i to node j. This chapter focuses on EH-WSNs and retains the symbols introduced by Bogliolo *et al.* (Bogliolo et al., 2010). Each node (say, v) is annotated with two variables: $P(v)$, which represents the environmental power available at that node, and $CPU(v)$, which represents its computational power expressed as the number of packets that can be processed in a time unit. Similarly, each edge (say, e), is annotated with variable $C(e)$, which represents the capacity (or bandwidth) of the link, and

variable $E(v,e)$, which represents the energy required at node v to process (receive or generate) a data packet and to forward it through its outgoing edge e.

The maximum number of packets that can be steadily sent across e in a time unit (denoted by $cap(e)$) is limited by its bandwidth ($C(e)$), by the processing speed of the source node ($CPU(v)$), and by the ratio between the environmental power available at the node and the energy needed to process and send a packet across e ($P(v)/E(v,e)$). In fact, the ratio between the energy needed to process a packet and the power harvested from the environment represents the time required to recharge the energy buffer in order to be ready to process a new packet. The inverse ratio is an upper bound for the sustainable packet rate. In symbols:

$$F(e) \leq cap(e) = \min\{C(e), CPU(v), \frac{P(v)}{E(v,e)}\} \tag{1}$$

where $F(e)$ is the packet flow over edge e. Since $cap(e)$ is an upper bound for $F(e)$, it can be treated as a link capacity that summarizes all the constraints applied to the edge, suggesting that the overall sensor network can be modeled as a *flow network* (Ford & Fulkerson, 1962).

Each node, however, usually has multiple outgoing edges that share the same power and computational budget, so that capacity constraints cannot be independently associated with the edges without taking into account the additional constraints imposed to their source nodes, represented by the following equations:

$$\sum_{e\_exiting\_from\_v} F(e) \leq CPU(v) \tag{2}$$

$$\sum_{e\_exiting\_from\_v} F(e)E(v,e) \leq P(v) \tag{3}$$

If the transmission power is not dynamically adapted to the actual length of the wireless link (Wang & Sodini, 2006), the energy spent at node v to process a packet can be regarded as a property of the node (denoted by $E(v)$) independent of the outgoing edge of choice. In this case, which is typical of most real-world WSNs, the constraints imposed by Equations 2 and 3 can be suitably expressed as capacity constraints (denoted by $cap(v)$) applied to the packet flow across node v (denoted by $F(v)$). In symbols:

$$F(v) = \sum_{e\_exiting\_from\_v} F(e) \tag{4}$$

$$F(v) \leq cap(v) = \min\{CPU(v), \frac{P(v)}{E(v)}\} \tag{5}$$

Node-constrained flow networks can be easily transformed into equivalent edge-constrained flow networks by splitting each original constrained node (v) into an *input sub-node* (destination of all incoming edges) and an *output sub-node* (source of all outgoing edges) connected by an internal (virtual) edge with capacity $cap(v)$ (Ford & Fulkerson, 1962). All other edges, representing the actual links among the nodes, maintain their original capacities according to Equation 1. The result is an edge-constrained flow network which retains all

the constraints imposed to the sensor network, allowing us to handle EH-WSNs within the framework of flow networks.

When node constraints cannot be expressed as cumulative flow limitations independent of the incoming or outgoing edges, however, the network cannot be transformed into an equivalent edge-constrained flow network. Any directed graph with arbitrary flow limitations possibly imposed at both edges and nodes, will be hereafter called *generalized flow network*.

For the sake of explanation we consider sensor networks made of 4 types of nodes: *sensors*, which are equipped with transducers that make them able to sense the environment and to generate data packets to be sent to a collection point, *sinks*, which generate control packets and collect data packets, *routers*, which relay packets according to a given routing algorithm, and *sensor-routers*, which exhibit the behavior of both sensors and routers. Without loss of generality, in the following we consider a sensor network with only one sink. Generalization to multi-sink networks can be simply obtained by adding a dummy sink connected at no cost with all the actual sinks (modeled as routers).

Figure 1 shows a hierarchical sensor network (Iwanicki & van Steen, 2009) of 64 sensors (thin circles), 16 routers (thick circles), and 1 sink (square) which will be used throughout the rest of this chapter to illustrate the routing strategy and to test its performance. Sensors and routers are uniformly distributed over a square 10x10 region, with the sink in the middle. The communication range of each node is equal to the minimum diagonal distance between the routers (edges are not represented for the sake of simplicity). Shading is used to highlight the sensors that need to be sampled according to a given monitoring task. The case of Figure 1 refers to a monitoring task involving only the 4 sensor nodes in the upper-left corner of the coverage area.

Definition 1. Given a WSN, the environmental conditions (expressed by the distribution of environmental power available at each node), and a monitoring task, the *maximum sustainable workload* (MSW) for the network is the maxflow from the sampled sensors to the sink in the corresponding flow network.

Since maxflow is defined from a single source to a single destination (Ford & Fulkerson, 1962), if there are multiple sensors that generate packets simultaneously, a dummy source node with cost-less links to the actual sources needs to be added to the model. The maxflow from the dummy source to the sink represents the global MSW.

The MSW can be determined in polynomial time by solving an instance of the *maximum-flow problem* within the theoretical framework of flow networks (Ford & Fulkerson, 1962).

If the transmission power is tuned to the length (and quality) of the links, the energy per packet depends on the outgoing edge, so that Equations 2 and 3 cannot reduce to Equation 5, node constraints cannot be transformed into equivalent edge constraints, and classical maxflow algorithms cannot be applied. Nevertheless, the network is still a generalized flow network, the maxflow of which represents the MSW of the corresponding WSN.

The theory presented in this chapter is not aimed at determining the MSW of a WSN. Rather, it is aimed at designing a routing algorithm able to route any sustainable workload, including

the theoretical maximum. Hence, we are interested in the value of MSW at the only purpose of testing the routing algorithm under worst case operating conditions.

If the monitoring task consists of sampling a given subset of the sensor nodes, the MSW is directly related to the *maximum sustainable sampling rate* (MSSR) at which all target nodes can be simultaneously sampled by the sink without violating power, bandwidth, and resource constraints (Lattanzi et al., 2007).

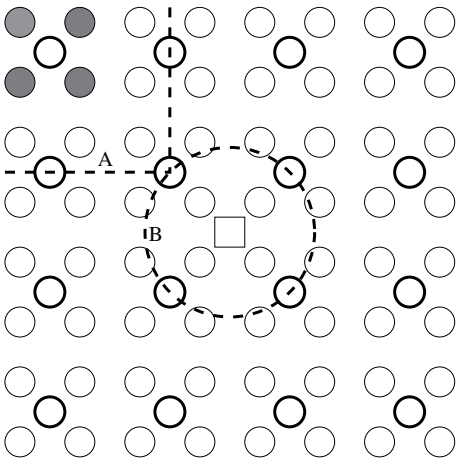


Fig. 1. Hierarchical network used as a case study.

3. Self-adapting maxflow routing algorithm

Capacity constraints, path capacities, and bandwidth requirements can be expressed in terms of packets per time unit. Since dynamic routing strategies can take different decisions for routing each packet, the routing algorithm can be developed by looking at packets rather than at overall flows. The capacity constraints imposed at a given node (edge) at the beginning of a time unit represent the actual number of packets that can be processed by that node (routed across that edge) in the time unit. Whenever the node (edge) is traversed by a packet, its *residual capacity* (which represents its capability of handling other packets in the same time unit) is decreased because of the energy, CPU, and bandwidth spent to process that packet.

Given a generalized flow network with vertices V and edges E, a path of length n from a source node s to a destination node d is a sequence of nodes $\mathcal{P} = (v_0, v_1, ..., v_{n-1})$ such that $v_0 = s$, $v_{n-1} = d$, and $e_{v_{i-1},v_i} \in E$ for each $i \in [1, n-1]$. We call *path capacity* of $\mathcal{P}$, denoted by $cap(\mathcal{P})$, the maximum number of packets per time unit that can be routed across the path without violating any node or edge constraint. We call *point-to-point flow* the flow of packets routed from one single source to one single destination, regardless of the path they follow.

Referring to a path $\mathcal{P}$ and to a time unit t, the *nominal path capacity* of $\mathcal{P}$ at time t is the capacity of the path computed at the beginning of the time unit by assuming that all the resources along the path are entirely assigned to $\mathcal{P}$ for the whole time unit. In practice, it corresponds to the minimum of the capacities of the nodes and edges belonging to the path, as computed at the

beginning of the time unit. The *residual path capacity* of $\mathcal{P}$ at time $t + \tau$, on the contrary, is the path capacity re-computed at time $t + \tau$ by taking into account the resources consumed up to that time by the packets processed since the beginning of the time unit.

The *self-adapting maxflow* (SAMF) routing algorithm proposed by Bogliolo *et al.* (Bogliolo et al., 2010) implements a simple *greedy* strategy that can be described as follows: *always route packets across the path with maximum residual capacity to the sink*.

According to this strategy, the residual path capacity is used as a routing metric. More precisely, the metric used at node v to evaluate its outgoing edge e is the maximum of the residual capacities of all the paths leading from v to the sink through edge e. The minimum number of hops can be used as a second criterion to choose among edges with the same residual path capacity.

The complexity of the routing algorithm is hidden behind the real-time computation of residual path capacities, which are possibly affected by any routed packet and by any change in the constraints imposed to the nodes and to the edges encountered along the path. In principle, in fact, routing metrics should be recomputed at each node (and possibly diffused) whenever a data packet is processed or an environmental change is detected.

In order to reduce the control overhead of real-time computation of residual path capacities, routing metrics can be kept unchanged for a given time period (called *epoch*) regardless of traffic conditions and environmental changes, and recomputed only at the beginning of a new epoch. In this way, all the packets processed by a node (say v) in a given epoch are routed along the same path, which is the one with the highest nominal capacity as computed at the beginning of that epoch. Residual capacities are computed at the end of the epoch by subtracting from nominal capacities the cost of all the packets routed in that epoch (in terms of energy, CPU, and bandwidth).

The lack of feedback on the effects of the routing decisions taken within the same epoch may cause the nodes to keep routing packets along saturated paths, leading to negative residual capacities at the end of the epoch. Negative residual capacities (hereafter called *capacity debts*) represent temporary violations of some of the constraints. Depending on the nature of the constraints (power, CPU, bandwidth) the excess flow that causes a capacity debt can be interpreted either as the amount of packets enqueued at some node waiting for the physical resources (bandwidth or CPU) needed to process them, or as the extra energy taken at some node from an auxiliary battery that needs to be recharged in the next epoch. In any case, capacity debts need to be compensated in subsequent epochs. This is done by subtracting the debts from the corresponding nominal capacities before computing nominal path capacities at the beginning of next epoch. **Example 1.** Consider the network of Figure 1 with the same constraints (namely, $cap(v) = 200$) imposed to all sensors and routers, and with no edge constraints. The effect of a SAMF routing strategy are shown in Figure 2, where sensor nodes and edges not involved in the monitoring task are not represented for the sake of simplicity. Intuitively, the maxflow is 600, corresponding to a MSSR of 150 packets per sensor per unit. In fact, all data packets need to be routed across cut A (shown in Figure 1), which contains only 3 routers with an overall capacity of 200x3=600 packets per time unit. An optimal flow distribution is shown in Figure 2.d, where the thickness of each edge represents the flow it sustains: 50 packets per time unit for the thin lines, 200 packets per time unit for the thick

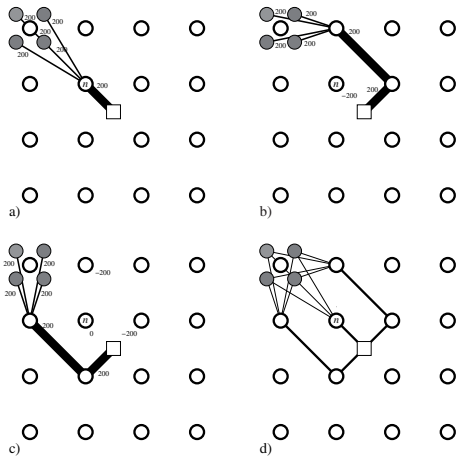


Fig. 2. d) Maxflow of the example network, obtained by averaging the flows over three epochs: a), b), and c).

ones. Figures 2.a, 2.b, and 2.c show the flows allocated by the SAMF routing strategy in three subsequent epochs (of one time unit each) when the 4 sensors of interest are sampled at the MSSR (namely, 150 packets per time unit). Nominal node capacities at the beginning of each epoch are annotated in the graphs in order to point out the effects of over-allocation. In the first epoch all the paths to the sink exhibit the same capacity, so that the path length is used to choose the best path. Since the routing metric is not updated during the first epoch, all data packets (namely, 600) are pushed along the shortest path from the upper corner to the sink, which could sustain only 200. The residual capacity of node n at the end of the first epoch is -400. The capacity debt of 400 packets is then subtracted from the nominal capacity of node n (200) at the beginning of the second epoch, that becomes -200. This negative value imposes to the algorithm the choice of a different path. Since the capacity debt of the shortest path is completely compensated at the end of the third epoch, the entire routing strategy can be periodically applied every three epochs. The optimal maxflow solution of Figure 2.d can be obtained by averaging the flows allocated by the self-adapting algorithm in the three epochs shown in the figure.

We say that a routing strategy *converges* if it can run forever causing only finite capacity debts. The convergence property of the SAMF routing strategy is stated by the following theorem.

Theorem 1. Given an autonomous WSN with power, bandwidth, and resource constraints expressed by Equations 1 and 5, the SAMF routing strategy converges for any sustainable workload.

Proof. Assume, by contradiction, that a sustainable workload is applied to the network but the strategy does not converge, so that there is at least one edge (or one node) with a capacity debt which keeps increasing and a residual capacity which decreases accordingly. Without loss of generality, assume that edge e, from node i to node j, has the lowest residual capacity at the end of time epoch h, denoted by $cap(e)^{(h)}$. If the routing strategy does not converge, for each epoch h there is an epoch $k > h$ such that the residual capacity of e at the end of k is lower

than that at the end of h. In symbols:

$$cap(e)^{(k)} < cap(e)^{(h)}$$

The decrease of the residual capacity in e from epoch h to epoch k means that the average flow routed across e in that interval of time exceeds the nominal capacity of e. We distinguish two different cases.

In the first case all the point-to-point flows routed across e have no alternate paths to the sink. Hence, if the sum of the flows exceeds the capacity of e, then the workload is not sustainable. A contradiction.

In the second case the edge is also used to route at least one multi-path point-to-point flow. Since the routing metric is based on residual path capacities, a path including edge e can be taken iff all alternate paths have lower residual capacities. Since e was the edge with lowest residual capacity at epoch h, then it can be taken at some epoch l from h to k iff the capacities of all alternate paths have become lower in the mean time. But this may happen iff the average flows routed across all alternate paths have exceeded their nominal capacities, meaning that the workload is not sustainable for the network. A contradiction.

Theorem 1 demonstrates that SAMF routing is able to route any sustainable workload under power, bandwidth, and resource constraints. Moreover, it has the inherent capability of adapting to environmental changes expressed as time-varying constraints.

It is worth noticing, however, that the theory exposed so far doesn't take into account the control traffic overhead required to recompute routing metrics at the beginning of each time epoch, nor the non-idealities of the wireless channels used for communication. The impact of traffic overhead and non idealities will be extensively discussed in Section 5.

4. Design and simulation tools

The theoretical framework described in Section 2 and the routing algorithm outlined in Section 3 provide the basis for the development of design methodologies for WSNs with unlimited lifetime.

In fact, the algorithmic solutions to the maximum-flow problem (Ford & Fulkerson, 1962) enable the evaluation of the MSW of a given WSN under specific environmental conditions and monitoring tasks. Techniques for the computation of the *maximum energetically sustainable workload* (MESW) were proposed by Bogliolo *et al.* (Bogliolo et al., 2006) for different monitoring tasks, including *selective monitoring* (i.e., sampling of a single node at the maximum sustainable rate), *non-uniform monitoring* (i.e., sampling a cluster of sensor nodes to generate the maximum overall traffic), and *uniform monitoring* (i.e., sampling a cluster of nodes at the maximum sustainable common rate). While the first two tasks can be directly solved as instances of maxflow, uniform monitoring requires an iterative approach which makes use of maxflow in the inner loop (Bogliolo et al., 2006). The same algorithms originally developed to determine MESW, can be applied to determine the more general MSW, which also takes into account CPU and capacity constraints.

The capability of evaluating the MSW can be used, in its turn, to drive the design of sustainable routing algorithms and the deployment of energy-aware WSNs. The concept of *MESW optimality* was introduced to this purpose (Lattanzi et al., 2007). A MESW-optimal non-deterministic routing algorithm can be directly derived from the solution of maxflow: each edge can be chosen by the algorithm with a probability proportional to the amount of flow across that edge in the solution of maxflow. Edge probabilities can be stored as routing tables at each node and used at run time to take pseudo-random decisions. A static version of this algorithm was originally implemented on *Tmote Sky* nodes (Klopfenstein et al., 2007). The SAMF routing algorithm outlined in Section 3 achieves the same MESW optimality while dynamically adapting to time-varying conditions.

The self-adapting capabilities of SAMF routing, together with the proof of optimality given by Theorem 1, provide a practical mean for overcoming the limitation of maxflow algorithms which cannot be applied to generalized flow networks subject to node constraints that cannot be transformed into edge constraints. First of all, thanks to the proved optimality, SAMF algorithm can be directly applied to any WSN without further optimization steps. Second, a WSN running the SAMF algorithm can be used to check the sustainability of a given workload. Iterative approaches have been developed on top of a simulation model of SAMF algorithm to determine the MSW of generalized flow networks (Seraghiti et al., 2008), as detailed in Subsection 4.2.

Finally, accurate network simulation models are required to investigate the practical applicability of MSW-optimal algorithms by evaluating the additional features of practical interest (such as control traffic overhead, number of hops, convergence speed, maximum buffer size, and scalability) and the effects of real-world non-idealities (such as communication time, radio broadcasting, radio-channel contention, and packet loss). The inherent features of the SAMF algorithm were evaluated by running extensive experiments with the simulation model implemented on top of OMNeT++, a discrete-event, open-source, modular network simulator (Bogliolo et al., 2010). Network components were written in C++ and composed using a high-level network description language called *NED*. The evaluation of the impact of non-idealities has prompted for the development of a more realistic simulation model which is presented in the following subsection.

4.1 SAMF simulator

The simulator presented in this subsection has been conceived to allow the designer to directly execute the Java bytecode written for the target sensor nodes (namely, `Sentilla JCreate` or other sensor nodes running an embedded JVM) while simulating their power consumption and the effects of realistic channel models. Since non-idealities can be selectively enabled or turned off, the simulator bridges the gap between theory and practice in that it can be used both to reproduce the theoretical results (when launched with ideal channel models) and to simulate real-world conditions (when launched with channel and energy models characterized on the field).

The simulator has been developed on top of the `SimJava` framework (Kreutzer et al., 1997), an event-driven multi-threaded simulator which handles concurrent entities (`Sim_entity` objects) running on separate threads and communicating through uni-directional channels

established between their ports. The multi-threaded nature of `SimJava` has been deeply exploited to simulate the concurrency among the nodes and among the tasks executed at each node. Each sensor node has been implemented by means of two separate instances of `Sim_entity`: a `SensorSw` executing the bytecode to be loaded on the target sensor node and a `SensorHw` catching low level calls and emulating the behavior of the hardware, including the routing protocol. In case of multiple applications running on the same node, each of them is assigned an independent instance of `SensorSw`.

In order to make it possible for the simulator to run the same bytecode compiled for the target Sentilla nodes, the libraries included in the Sentilla framework have been re-implemented by exposing the same API while allowing the `SensorHw` to: catch the appropriate calls, exhibit the required behavior, emulate hardware devices (including LED, radio devices, and sensors), and enable instrumentation.

Modeling wireless (i.e., broadcast) communication channels on top of SimJava has required the development of new class (i.e., `Network`) which extends `Sim_Entity` and works as a network dispatcher. In practice, it is connected to all sensor nodes through bi-directional virtual channels (with no correspondences with real-world channels) which allow the network to take full control of the actual topology of the WSN, to catch all communication events, to implement channel models, to inject non-idealities, and to deliver packets.

The `Network` class does not contain the channel model. Rather, the model is specified in a separate object, called `ChannelBehavior`, which is loaded by the constructor. The level of realism of the simulation can be tuned by changing the channel behavior.

4.2 Test of sustainability

According to Theorem 1, the SAMF routing strategy implemented by the simulation model outlined so far is guaranteed to converge for any sustainable workload. Hence, simulation stability can be used as a proof of sustainability for the workload applied to the network. The instability of the simulation can be detected by checking for capacity debts that keep increasing over time. The simulation is considered to be convergent if it lasts for a "long-enough" number of epochs without causing any instability.

Given the simulation model of a WSN, the simulator can be viewed as a function which takes in input two parameters: the sampling rate (SR) to be applied to the target sensors and the simulation length (Nepochs). When the function is invoked the simulation is launched and the function returns 1 in case of instability or 0 in case of normal termination. The function can then be used within the inner loop of a bisection-search algorithm in order to estimate the *maximum sustainable sampling rate* of a uniform monitoring task. We denote by $\tilde{MSSR}$ the estimated value of MSSR, which suffers from two sources of approximation: the limited number of iterations in the bisection-search algorithm (each iteration adds a binary digit to the precision) and the limited length of simulation runs (which avoids the detection of instabilities that would show up after the end of simulation). As a result, $\tilde{MSSR}$ overestimates the actual value of MSSR with a precision given by $n \log_{10} 2$, where n is the number of bisection iterations. For instance, if the bisection algorithm iterates $n = 10$ times, then the precision of the estimator is 3 digits, corresponding to a maximum overestimation of 1 per mil.

5. Simulation results

This section makes use of the simulator described in Section 4.1 in order to test and discuss the performance of the SAMF algorithm and its sensitivity to design parameters and real-world operating conditions. The parametric nature of the simulator and its capability of using different channel models are used hereafter to validate simulation results against their theoretical counterparts (Subsection 5.1), to incrementally add non-idealities to make simulation more realistic (Subsection 5.2), and to conduct a sensitivity analysis on a large set of Monte Carlo experiments (Subsection 5.3).

5.1 Validation

Validation was performed by running the simulation-based iterative procedure outlined in Section 4.2 with ideal channel models in order to determine the value of $M\tilde{S}SR$ (i.e., the estimated value of the maximum sustainable sampling rate). WSNs without edge-dependent node constraints were used to this purpose in order to apply classical maxflow algorithms (Ford & Fulkerson, 1962) to compute the theoretical value of $MSSR$ on the equivalent flow network. The ratio between $M\tilde{S}SR$ and $MSSR$ represents the so-called *optimality ratio*, which was originally introduced to express the optimality of routing algorithms (Lattanzi et al., 2007). When the optimality of the algorithm under study is known a priori, as in case of the SAMF algorithm applied in ideal conditions, the ratio provides a measure of the accuracy of the simulator.

The validation procedure was applied to the example of Figure 1 by running the bisection search algorithm for 20 iterations with simulations lasting for 1,000 epochs of 50 time units each. The optimality ratio obtained whitouth taking into account the control traffic overhead (in order to make simulation results directly comparable with the theoretical optimum) was 0.990, while the value achieved with control traffic overhead was 0.973, demonstrating both the accuracy of the simulator and the small overhead of control packets in the experimental settings adopted.

5.2 Effects of non-idealities

Specific models were implemented on top of the simulator described in Section 4.1 to investigate the effects of the following non-idealities:

- *communication delay*, which represents the propagation time across the channel;
- *de-synchronization*, which is modeled as a boot time randomly generated for each node;
- *transmission time*, which represents the time required by the transmitter node to send the whole packet across the channel;
- *channel collision*, which avoids a destination node to properly receive two packets with overlapping transmission times;
- *packet loss*, which represents the probability of discarding a packet because of the bit error rate of the channel;
- *reception energy*, which represents the energy spent to receive a packet at each node which is in the range of the transmitter, independently of the destination address.

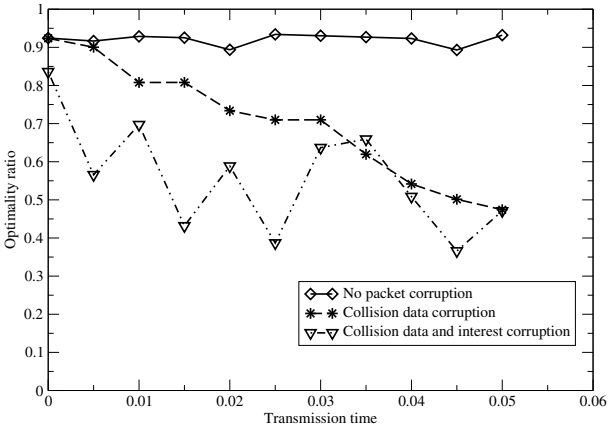


Fig. 3. Optimality ratio as a function of transmission time.

The results presented hereafter refer to the simple WSN of Figure 1 simulated with the same parameters used for validation. It is worth noting that the MSW is measured as the overall number of packets received at the sink in a time unit when sampling at the same rate the 4 sensor nodes placed in the upper-left corner of the coverage area.

5.2.1 Timings

The introduction of communication delay and de-synchronization produced negligible effects on the optimality ratio, demonstrating the robustness of the SAMF algorithm with respect to timing uncertainties and misalignments. Simulating a non-null transmission time has the only effect of keeping the channel busy during transmission, imposing an upper bound to the packet rate. If the simulated production rate does not exceed the physical upper bound of the channel, transmission time does not impact simulation results. This is shown by the solid curve of Figure 3, which plots the optimality ratio as a function of transmission time.

5.2.2 Channel contention

The time spent to send a packet has a sizeable impact on performance when channel collision is simulated. In this case, in fact, the channel cannot be simultaneously used by neighboring nodes, or otherwise collisions would cause the corruption of the packets. Channel sensing mechanisms with pseudorandom retry time were simulated in order to manage channel contention. Although this simple mechanism does not avoid collisions occurring at a destination node receiving simultaneous packets from two or more non adjacent nodes, it affects the optimality ratio for three main reasons: first, because of the loss of collided packets which do not reach the sink, second, because of the induced correlation between the activity of adjacent nodes, third, because of the reduced path diversity of the SAMF algorithm. In fact, since collided packets are discarded at the point of collision, they do not consume any energy along the rest of the path to the sink. Hence, the routing metrics based on residual energy induce the routing algorithm to keep sending packets across high-collision paths. The combined effect of these three phenomena is shown by the decreasing trend of the dashed

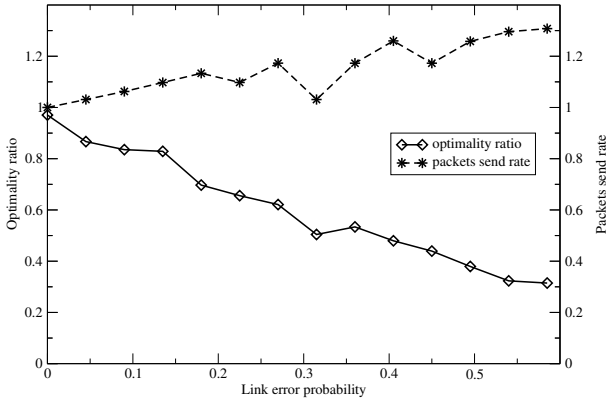


Fig. 4. Optimality ratio as a function of link error probability.

curve of Figure 3. The chaotic behavior of the dash-dotted curve is obtained by simulating also collisions occurring in the diffusion phase, thus causing a loss of interest packets which ultimately impacts the correctness of the SAMF algorithm.

5.2.3 Link quality

Figure 4 plots the effects of packet loss due to link error probability independent of packet collisions. Although the MSW decreases for increasing values of the link error probability, the effect is much easier to explain than that caused by packet collisions. In this case, in fact, packet loss is independent of traffic congestion and it does not induce any correlation between adjacent nodes. Hence, the loss of optimality is only caused by the reduced percentage of packets which reach the sink. The dashed curve in Figure 4 shows the increased sampling rate imposed to the sensor nodes in order to compensate for the loss of packets. A deeper analysis of simulation results highlights that the higher the link error rate the shorter the paths used on average to route the packets. In fact, since the errors are independently injected at each link, the probability of losing a packet along a path increases with the path length.

5.2.4 Reception energy

Wireless transmission is based on radio broadcasting. This means that each node receives all the packets transmitted by its adjacent nodes, even if it is not along the path selected by the routing algorithm. In case of point-to-point transmission across a wireless link, the packet is discarded by all the receiving nodes but the destination one. However, some energy is spent at each node to receive the packet and check its destination address before taking the decision to discard it. The energy wasted to listen to a broadcast channel has a deep impact on the energy efficiency of a WSN. This phenomenon is often neglected by energy-aware routing algorithms, which are mainly focused on transmission/processing energy spent by nodes which lay along the routing path. Figure 5 plots the optimality ratio as a function of the ratio between the reception (RX) and transmission (TX) energy of the sensor nodes. It is worth noticing that when RX energy is about one tenth of TX energy, the MSW reduces to 50% of the theoretical optimum. When RX energy equals TX energy (which is a typical situation) the optimality

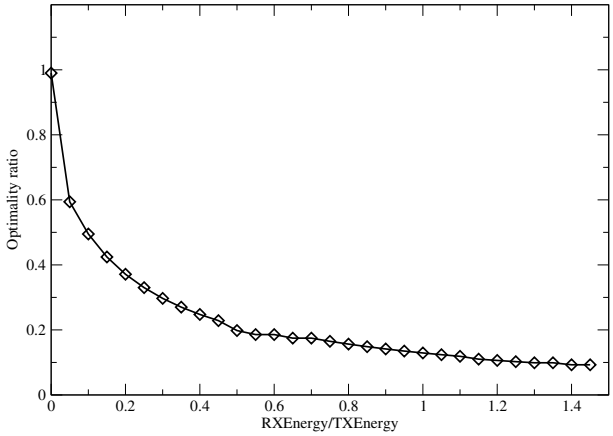


Fig. 5. Optimality ratio as a function of the ratio between RX and TX energy.

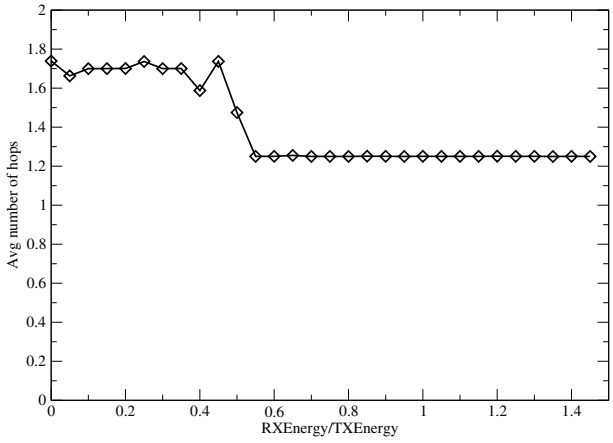


Fig. 6. Path length as a function of the ratio between RX and TX energy.

ratio reduces below 20%. Figure 6 reports the average number of hops from the source sensor nodes to the sink, as a function of RX energy. Since the packets routed along the best path also cause a sizeable waste of energy in the neighboring paths, RX energy significantly reduces the degrees of freedom available to the SAMF algorithm. For the case study of Figure 1, when RX energy accounts for more than 60% of TX energy, the crosstalk effect avoids the SAMF algorithm to take advantage of path diversity and the routing strategy resorts to minimum path.

5.3 Monte Carlo experiments

Monte Carlo simulations were conducted to perform a sensitivity analysis by means of pseudo-random sampling in a neighborhood of a given point in the design space. To this

purpose, we used parametrized randomly generated WSNs composed of nodes scattered in a square region with a sink in the middle. All the nodes (but the sink) have sensing and routing capabilities and are involved on a uniform-sampling task targeting the whole area.

The following simulation parameters were used as independent variables: the number of sensor nodes (*NSensors*), the transmission range of each node (*TxRange*), the energy spent at each node to transmit and process a packet (*TxEnergy*), the energy spent at each node to receive a packet (*RxEnergy*), the environmental power available at each node (*Power*), the length of the time epochs adopted for recomputing routing metrics (*Epoch length*), the capacity of the energy buffer installed at each node (*Energy buffer*), the transmission time (*TxTime*), the propagation delay (*Link delay*), and the link error probability (*Link err. prob.*).

The sensitivity analysis was conducted for the following (dependent) parameters of interest: the estimated values of MSW ($\tilde{MSW}$), the maximum capacity debt observed during the whole simulation (*MDebt*), the average path length (*Path l.*), the control traffic overhead (*Overhead*), the total amount of data packets routed (*Routed data*), and the total amount of collisions occurred during simulation (*Collision*).

The analysis was conducted on a sample of 1,000 points in parameter space. Each sampling point corresponds to a configuration of the independent variables uniformly taken from the ranges reported in Table 1. For each configuration, 3 random *trials* were performed using different seeds, resulting in 3,000 runs of the simulation-based bisection search of $\tilde{MSW}$.

Table 1 summarizes the results of the sensitivity analysis. Rows and columns are associated with independent variables and dependent parameters, respectively. The second column reports the sampled value range of each independent variable, while the second and third rows report the sample average and standard deviation of the dependent parameters. All other entries of Table 1 report the correlation coefficients between independent and dependent variables computed on the results of the 3,000 Monte Carlo experiments. The most significant correlations (with absolute value greater or equal than 0.2) are highlighted in boldface and discussed in the following, column by column.

- $\tilde{MSW}$ is negatively affected by the number of sensor nodes (*NSensors*), which reduces the sustainable sampling rate of each sensor because of the limited routing capabilities of the network, and positively affected by the transmission range (*TxRange*), which increases the number of paths available to route data packets. Interestingly enough, the negative impact of *RxEnergy* on $\tilde{MSW}$ is much higher than that of *TxEnergy*, because of the energy waste induced in the neighborhood of the routing path discussed in Section 5.2. As expected, a high correlation coefficient is observed between $\tilde{MSW}$ and environmental power (*Power*), while the link error probability negatively affects the maximum sustainable workload.
- *Mdebt* is mainly affected by the environmental power because of its high correlation with $\tilde{MSW}$. In fact, the debt is caused by the excess of packets routed across a saturated path in a time epoch because of the lack of feedback on the residual path capacity. The higher the sampling rate, the higher the debt that can be reached during simulation.
- *Path length* increases with the overall flow, which depends, in its turn, from the environmental power (*Power*). In fact, the larger the flow the larger the number of paths (possibly longer than the minimum one) that need to be used by the routing

		$M\tilde{S}W$	*MDebt*	*Path l.*	*Overhead*	*Routed data*	*Collisions*
	Average	0.18	25,095	2.688	20,427	24,934	19,729
	STD	0.09	11,753	1.302	11,003	10,946	28,609
Edge	[100, 200]	0.181	-0.027	0.117	-0.028	0.052	-0.096
NSensors	[15, 25]	**-0.218**	0.040	-0.084	**0.541**	-0.069	0.187
TxRange	[50, 100]	**0.399**	0.167	**-0.779**	**0.243**	**-0.725**	-0.062
TxEnergy	[100, 200]	-0.140	-0.048	-0.005	0.036	0.011	-0.035
RxEnergy	[100, 200]	**-0.405**	0.011	0.026	0.031	0.050	**-0.308**
Power	[500, 5000]	**0.433**	**0.492**	**0.239**	-0.022	-0.039	**0.265**
Epoch length	[50, 100]	-0.030	-0.004	-0.012	**-0.688**	-0.011	-0.049
Energy buffer	[50k, 500k]	0.024	0.004	-0.031	0.002	0.046	0.011
TxTime	[0.0, 0.05]	-0.012	-0.067	0.065	-0.097	0.047	**0.672**
Link delay	[0.0, 0.1]	0.004	0.102	-0.043	-0.082	-0.073	0.133
Link err. prob.	[0.0, 0.5]	**-0.374**	0.056	**-0.271**	0.147	**-0.211**	0.087

Table 1. Results of the sensitivity analysis, expressed by the correlation coefficients between independent variables (rows) and dependent parameters (columns). Significant correlations are highlighted in bold.

strategy. As expected, the average path reduces for longer transmission ranges (*TxRange*). Interestingly, path length decreases for higher values of link error probability (*Link err. prob.*). This is due to the fact that statistics are computed on packets received by the sink, and packets routed across longer paths have a lower probability of reaching the sink.

- *Overhead* is positively affected by the number of nodes (*NSensors*) and by the transmission range (*TxRange*). In fact, the number of Interest messages received and sent by each sensor depends on the number of incoming and outgoing edges, respectively. Both the number of nodes and the transmission range positively affect the degree of connectivity, causing a larger overhead. As already discussed, the overhead reduces when the epoch length increases.
- *Routed data* is negatively affected by the transmission range (*TxRange*), which decreases the number of hops needed to reach the sink, and by link error probability (*Link err. prob.*), which favors shorter paths.
- *Collisions* is positively affected by *Power* and negatively affected by *RxEnergy*. Both correlations can be explained by looking at the maximum sustainable workload. In fact, the collision probability increases with traffic. Finally, collisions are strongly affected by the transmission time, which increases the risk of overlapping of the transmission intervals of two or more independent packets sent to the same node.

6. Conclusions

Energy harvesting (EH) techniques enable the development of wireless sensor networks (WSNs) with unlimited lifetime. This attractive perspective prompts for a paradigm shift in the design and management of EH-WSNs.

This chapter has provided a thorough overview of the results recently achieved in the design of EH-WSN within the framework of generalized flow networks, including the

network model, the concept of maximum sustainable workload (MSW), and the self-adapting maxflow routing algorithm (SAMF). In particular, the SAMF algorithm is able to route any theoretically sustainable workload while autonomously adapting to time varying environmental conditions. It has been shown that the MSW is a suitable design metric for EH-WSNs with unlimited lifetime, while the SAMF algorithm can be used within the inner loop of bisection search algorithms to estimate the MSW for generalized flow networks which cannot be handled by traditional maxflow algorithms.

Finally, a new simulator has been developed to evaluate the practical applicability of the theoretical results. The simulator has been validated by reproducing theoretical results under ideal operating conditions, and then used to inject real-world non idealities, including propagation delay, de-synchronization, channel contention, packet loss, and reception energy. The sensitivity analysis conducted on a large set of Monte Carlo simulation experiments allows the designer to figure out the performance of the SAMF algorithm in many different real-world scenarios. In particular, it has been pointed out that the maximum sustainable workload is highly affected by the reception energy which is spent at each node to receive broadcast packets independently of their destination address. The reception energy wasted by nodes which are not along the routing path is usually neglected by energy-aware routing algorithms since it is a side effect which is not captured by routing metrics.

Future directions in the field of WSNs with unlimited lifetime include: modeling reception energy within the framework of generalized flow networks, developing sensor nodes able to reduce the energy wasted in listening for packets addressed to other nodes, developing design tools using MSW as a metric, and implementing SAMF routing strategies in real-world WSNs.

7. References

Amirtharajah, R., Collier, J., Siebert, J., Zhou, B. & Chandrakasan, A. (2005). Dsps for energy harvesting sensors: applications and architectures, *IEEE Pervasive Computing* 4(3): 72–79.

Bogliolo, A., Delpriori, S., Lattanzi, E. & Seraghiti, A. (2010). Self-adapting maximum flow routing for autonomous wireless sensor networks, '*Cluster Computing* 14: 1–14.

Bogliolo, A., Lattanzi, E. & Acquaviva, A. (2006). Energetic sustainability of environmentally powered wireless sensor networks, *Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks*, pp. 149–152.

Chen, H.-H. & Yang, Y. (2007). Guest editorial: Network coverage and routing schemes for wireless sensor networks, *Elsevier Computer Communications* 30(14-15): 2697–2698.

Dressler, F. (2008). A study of self-organization mechanisms in ad hoc and sensor networks, *Elsevier Computer Communications* 31: 3018–3029.

Eu, Z. A., Tan, H.-P. & Seah, W. K. (2010). Opportunistic routing in wireless sensor networks powered by ambient energy harvesting, *Computer Networks* 54(17): 2943 – 2966.

Ford, L. R. & Fulkerson, D. R. (1962). *Flows in Networks*, Princeton University Press.

Hasenfratz, D., Meier, A., Moser, C., Chen, J.-J. & Thiele, L. (2010). Analysis, comparison, and optimization of routing protocols for energy harvesting wireless sensor networks, *Proceedings of Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 19–26.

Iwanicki, K. & van Steen, M. (2009). On hierarchical routing in wireless sensor networks, *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pp. 133–144.

Klopfenstein, L. C., Lattanzi, E. & Bogliolo, A. (2007). Implementing energetically sustainable routing algorithms for autonomous wsns, *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007)*, pp. 1–6.

Kreutzer, W., Hopkins, J. & van Mierlo, M. (1997). Simjava a framework for modeling queueing networks in java, *Proceedings of the 29th conference on Winter simulation*, pp. 483–488.

Kulkarni, R., Forster, A. & Venayagamoorthy, G. (2011). Computational intelligence in wireless sensor networks: A survey, *Communications Surveys and Tutorials, IEEE* 13(1): 68 – 96.

Lattanzi, E., Regini, E., Acquaviva, A. & Bogliolo, A. (2007). Energetic sustainability of routing algorithms for energy-harvesting wireless sensor networks, *Elsevier Computer Communications* 30(14-15): 2976–2986.

Levis, P., Culler, D., Gay, D., Madden, S., Patel, N., Polastre, J., Shenker, S., Szewczyk, R. & Woo, A. (2008). The emergence of a networking primitive in wireless sensor networks, *Communications of the ACM* 51(7): 99–106.

Li, C., Zhang, H., Hao, B. & Li, J. (2011). A survey on routing protocols for large-scale wireless sensor networks, *Sensors* 11(4): 3498–3526.

Lin, L., Shroff, N. B. & Srikant, R. (2007). Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources, *IEEE/ACM Trans. Netw.* 15: 1021–1034.

Mhatre, V. & Rosenberg, C. (2005). Energy and cost optimizations in wireless sensor networks: A survey, *in* A. Girard, B. Sanso & F. Vazquez-Abad (eds), *Performance Evaluation and Planning Methods for the Next Generation Internet*, Kluwer Academic Publishers.

Mottola, L. & Picco, G. P. (2011). Programming wireless sensor networks: Fundamental concepts and state of the art, *ACM Comput. Surv.* 43: 19:1–19:51.

Nallusamy, R. & Duraiswamy, K. (2011). Solar powered wireless sensor networks for environmental applications with energy efficient routing concepts: A review, *Information Technology Journal* pp. 1–10.

Seraghiti, A., Delpriori, S., Lattanzi, E. & Bogliolo, A. (2008). Self-adapting maxflow routing algorithm for wsns: practical issues and simulation-based assessment, *Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology*, pp. 688–693.

Shafiullah, G. M., Gyasi-Agyei, A. & Wolfs, P. J. (2008). *A Survey of Energy-Efficient and QoS-Aware Routing Protocols for Wireless Sensor Networks*, Springer Netherlands.

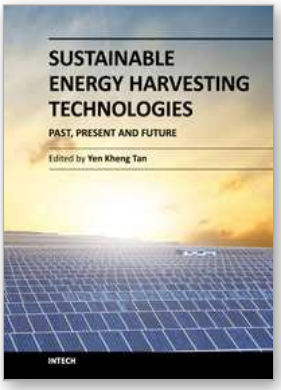
Sudevalayam, S. & Kulkarni, P. (2010). Energy harvesting sensor nodes: Survey and implications, *Communications Surveys and Tutorials, IEEE* pp. 1–19.

Wang, A. Y. & Sodini, C. G. (2006). On the energy efficiency of wireless transceivers, *Proc. of IEEE Conference on Communications*, pp. 3783–3788.

Yarvis, M. & Zorzi, M. (2008). Special issue on energy efficient design in wireless ad hoc and sensor networks, *Elsevier Ad Hoc Networks* .

Yick, J., Mukherjee, B. & Ghosal, D. (2008). Wireless sensor network survey, *Elsevier Computer Networks* 52: 2292–2330.

Zeng, K., Ren, K., Lou, W. & Moran, P. J. (2006). Energy-aware geographic routing in lossy wireless sensor networks with environmental energy supply, *Proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks*.



Sustainable Energy Harvesting Technologies - Past, Present and Future

Edited by Dr. Yen Kheng Tan

ISBN 978-953-307-438-2
Hard cover, 256 pages
Publisher InTech
Published online 22, December, 2011
Published in print edition December, 2011

In the early 21st century, research and development of sustainable energy harvesting (EH) technologies have started. Since then, many EH technologies have evolved, advanced and even been successfully developed into hardware prototypes for sustaining the operational lifetime of low?power electronic devices like mobile gadgets, smart wireless sensor networks, etc. Energy harvesting is a technology that harvests freely available renewable energy from the ambient environment to recharge or put used energy back into the energy storage devices without the hassle of disrupting or even discontinuing the normal operation of the specific application. With the prior knowledge and experience developed over a decade ago, progress of sustainable EH technologies research is still intact and ongoing. EH technologies are starting to mature and strong synergies are formulating with dedicate application areas. To move forward, now would be a good time to setup a review and brainstorm session to evaluate the past, investigate and think through the present and understand and plan for the future sustainable energy harvesting technologies.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Emanuele Lattanzi and Alessandro Bogliolo (2011). WSN Design for Unlimited Lifetime, Sustainable Energy Harvesting Technologies - Past, Present and Future, Dr. Yen Kheng Tan (Ed.), ISBN: 978-953-307-438-2, InTech, Available from: http://www.intechopen.com/books/sustainable-energy-harvesting-technologies-past-present-and-future/wsn-design-for-unlimited-lifetime

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821