A Distributed, Energy-aware, Utility-based Approach for Data Transport in Wireless Sensor Networks

Wei-Peng Chen, Jennifer C. Hou, Lui Sha and Marco Caccamo
Department of Computer Science
University of Illinois at Urbana-Champaign, Urbana, IL 61801
{*wchen3,jhou,lrs,mcaccamo*}*@cs.uiuc.edu*

Abstract*—Distinct from wireless ad hoc networks, wireless sensor networks are data-centric, application-oriented, collaborative, and energy-constrained in nature. In this paper, we formulate the problem of data transport in sensor networks as an optimization problem, with the objective of maximizing the amount of information (utility) collected at sinks (subscribers), subject to both the channel bandwidth and energy constraints. We then devise a distributed solution of the convex optimization problem, and explore in three directions. First, we devise a simple node capacity estimation method to on-line measure the node capacity (which changes with the traffic load and nodal distribution and is required in the optimization problem). Second, we linearize the energy constraint by properly setting the value of the system lifetime in advance and controlling the data rate of a node (and hence its total energy consumption rate) so as to sustain its battery lifetime longer than the specified lifetime. Finally, we incorporate the optimization results into routing so as to provide sensors with opportunities to select better routes. The simulation results show that the utility-based approach balances between system utility and system lifetime.

***Index Terms*—Utility-based, Pricing, Wireless sensor networks, System design , Simulations, Mathematical programming/optimization**

I. Introduction

Recent technological advances have led to the emergence of small, low-power devices that integrate sensors and actuators with limited on-board processing and wireless communication capabilities. Pervasive networks of such sensors and actuators open new vistas for constructing complex monitoring and control systems. Unlike traditional wired or wireless networks, sensor networks possess certain characteristics that warrant their treatment as a special class of ad hoc networks:

1) *Data-centric*: Sensor networks are largely data-centric, with the objective of delivering collected data in a timely fashion to destinations. Data that contains information of different qualities represents different values to destinations. As a result, the overall system objective is no longer to maximize the raw data throughput, but instead to maximize the amount of useful information carried to destinations.
2) *Application-oriented*: While traditional wired and wireless networks are expected to cast to a variety of applications, sensor networks are usually deployed to perform specific tasks. The specific algorithms/protocols and performance metrics used in sensor networks thus depend on the characteristics and requirements of applications. For instance, for mission-critical applications, it is very important to ensure the end-to-end latency be kept below certain threshold.
3) *Collaborative*: How nodes collaborate with each other to realize the global system objective outweighs the objective of achieving fairness of individual connections. This is in sharp contrast to conventional wired and wireless networks in which provisioning of fairness to users is an important design criterion.
4) *Energy-constrained*: As most of the low-power devices in sensor networks have limited battery life and replacing batteries on tens of thousands of these devices is infeasible, any protocol/algorithm that will be eventually deployed in sensor networks has to be energy aware.

As a result of the unique characteristics of sensor networks, conventional routing and flow control protocols that focus on maximizing raw data throughput and achieving fairness are no longer well suited for sensor networks. The architecture of directed diffusion [6] brings the notions of data-centric computing. However, it does not take into consideration of resource utilization, especially under the case of numerous queries from users simultaneously. A mechanism properly controlling the usage of scarce resource to

This work is supported in part by the MURI program N00014-01-0576, by the DARPA NEST program F33615-5-01-c-1905, by the NSF grant CCR-0325716, and by the NSF ANI under grant ANI-0221357.

deliver the most useful information to the sink is needed. Distributed, data-centric, utility based approaches that differentiate treatments of packets with respect to their different values and at the same time, take into account of both bandwidth usage and energy consumption are more adequate. Such mechanisms can solve simultaneously the problems of maximizing utility and mitigating congestion.

In this paper, we formulate the problem of data transport in sensor networks as an optimization problem, with the objective of maximizing the amount of information (utility) collected at sinks (subscribers), subject to the channel bandwidth and energy constraints. Note that channel bandwidth and energy constraints represent two of the most limited resource in wireless sensor networks. The problem formulation in our previous work [5] is a non-convex programming problem and a centralized approach is used to solve the optimization problem. As the centralized approach cannot quickly adapts to dynamic network changes, in this paper we devise a distributed, energy-aware, utility-based approach. The key to devising such a distributed solution is to transform the optimization problem to a convex programming problem. That is, the objective function has to be concave and the range of the feasible control variables is within a convex set. The advantage of a convex programming problem is that its dual problem can be solved in a distributed manner, as the control variables in the dual problem can be expressed in a separable form. Furthermore, there is no duality gap between the primal and dual problems.

There exist several challenges that we have to tackle in order to apply utility-based approaches to wireless sensor networks. The contributions of our paper can be summarized in three directions:

(I) Estimating node capacity in wireless multi-hop environments*:* As all the outgoing links from a node share the same channel in wireless multi-hop environments, capacity constraints should be imposed on the node capacity rather than on the link capacity. Constraints on the link capacity suffice only when a node is equipped with multiple transceivers operating at different channels or with directional antennas. Moreover, since the wireless channel is a shared medium, the channel capacity [1] is shared by all devices within the radio transmission range. As a result, different from the wireline environments in which the capacity of a link is given in its specification, the capacity of a node in wireless environments is no longer a constant but changes in different topologies and traffic status. It is non-trivial to determine the capacity of each node needed in the optimization problem. In this paper, we propose a capacity estimation method that adapts to the traffic load. The estimation is made based on (i) the measurements of average outgoing throughput and (ii) the feedback information of buffer overflows and unsuccessful transmissions. All the information can be obtained locally and the required computation is not intensive.

(II) Linearizing energy constraints*:* It is necessary to consider energy constraints because unattended sensors are equipped only with limited energy. If the system lifetime is considered as a control variable in the energy constraints such as in [5], the function that describes the energy constraint includes a product term of the system lifetime and the data rate of sources and is hence a non-convex function. As a non-convex programming problem cannot be solved in a distributed manner, we need to transform the energy constraint to a convex function. One method is to properly set the value of the system lifetime in advance and control the data rate of a node (and hence its total energy consumption rate) so as to sustain its battery lifetime longer than the specified lifetime. In this manner, the energy constraint becomes a linear function, as the system lifetime becomes a parameter but no longer a control variable. The price of the energy constraint is periodically adjusted according to the current energy consumption rate. Furthermore, both the capacity and energy constraints can be quantified in terms of the price based on how tight these constraints are. A more stringent constraint leads to a higher price.

(III) Integrating routing dynamics in the optimization*:* Like most existing flow control approaches, the proposed approach solves the problem of maximizing the amount of information delivered in two phases: a set of routes are determined in the first phase and then a convex programming problem is solved given the set of routes in the second phase. The resulting solution may not be optimal, but it has been shown in [13] that the problem of solving both routing and flow control simultaneously is NP-hard. As such, we incorporate the optimization results into routing, and propose a modified version of Ad hoc On Demand Distance Vector (AODV) routing protocol. The proposed routing protocol provides sensors with opportunities to select a route with a smaller price, and at the same time, improve the overall utility of the system. It is composed of three phases, route initialization (*RINT*), route request (*RREQ*) and route reply (*RREP*). In the first RINT phase, sink-trees constructed by sinks establish active connections from sensors to sinks and reduce the over-

[1] with the unit of bits/sec.

head of route discovery. As these sink-tree routes might incur high price due to the limited energy of certain intermediate nodes and/or congestion along these routes. The RREQ and RREP phases in the proposed protocol are then to find *alternate* routes with smaller costs. One issue has to be considered in particular: suppose a data flow f is to be switched to a new route, then the "losses" of data flows that are originally routed on this new route should be well compensated by the gain in switching the data flow f to this new route. By applying a price estimation method, we can reduce the impact of redirecting a data flow on the flows that are originally routed on the new route. In addition, the overheads incurred in the RREQ and RREP phases are controlled properly not to overload the networks.

Utility based approaches have been exploited in conventional wired networks (e.g., [7], [8]), cellular wireless networks (e.g., [11]), ad hoc networks (e.g., [10], [15]), and most recently sensor networks [2]. Kelly *et al.* [7] propose a pricing scheme to achieve weight proportional fair rate allocation for users in the wireline environment. The same problem considered in [7] is solved by Low *et al.* [8] differently such that the dual problem can be optimized in a distributed manner. Both Xue *et al.* [15] and Qiu *et al.* [10] extend Kelly's work [7] and consider the rate allocation problem in ad hoc networks. The major differences between Xue *et al.* [15] and Qiu *et al.* [10] lie in that (i) the former [15] uses the link capacity as the constraint of the channel capacity, while the latter [10] uses the node capacity as the constraint; and (ii) while the formulations in both work [15] and [7] divide the system problem into the user and network problems, the work reported in [10] incorporates the forwarding cost in the user optimization problem. None of the work in [7], [8], [15], [10] consider the energy constraints which we believe is one of the most important criteria in sensor networks.

Saraydar *et al.* [11] take a utility based approach to control transmission power in a decentralized manner in a multi-cell wireless data system. Recently Byers *et al.* [2] consider the optimization problem of maximizing the overall utility of sensor networks during the system lifetime, subject to an energy constraint. The energy constraint is, however, expressed as a high level cost, and does not differentiate power consumed in transmission, reception, and idle states. Chang *et al.* [3] devise a routing solution to maximize the system lifetime of sensor networks. As neither the link capacity nor the node capacity is considered in their work, the solution thus derived may not be feasible. Sadagopan *et al.* [12] solve a similar linear programming problem with an iterative approximation algorithm.

As compared with the aforementioned utility-based approaches, our proposed approach fixes problems of applying utility based approaches to wireless sensor networks in three directions: on-line estimation of node capacity (to be used in the capacity constraint of the optimization problem) in the multi-hop wireless environment, inclusion (and linearization) of energy constraints that relate the system lifetime to the data rate, and incorporation of optimization results in selecting routes that maximize the utility.

The rest of the paper is organized as follows. We define the problem and present a distributed flow control approach in Section II. Linearization of energy constraints, on-line estimation of the node capacity, and integration of dynamic routing with flow control are treated in Section III, IV, and V, respectively. Following that, we present the simulation results in Section VI. Finally we conclude the paper with a list of future research agendas in Section VII.

II. Problem Definition

In this section we briefly describe the scenario of applications in sensor networks to which the proposed utility-based approach is applied. Suppose a volcano observation system is to be implemented with a wireless sensor network. Sensors are placed near the volcanos to monitor their activities. The data rate of a sensor depends on the information of the activity a sensor observes. For instance, a low data rate is sufficient for the site of a dormant volcano. When a sensor detects an abnormal activity at a dormant volcano, it increases the data rate. If a volcano erupts, sensors then collect and transmit the status data with the highest data rate. The objective of the system is to deliver the largest amount of information (but not the largest amount of raw data) during the period of the system lifetime.

Before delving into the problem formulation, we state the assumptions made in this paper:

(A1) Spatial redundancy is not considered: we assume that the sensing data collected from sensors at different locations contributes additive utilities. In reality, surplus sensors may be deployed in the sensing area and the information collected by neighboring sensors may be redundant and correlated. Clustering techniques such as GAF [14] or SPAN [4] have been proposed to group sensors into clusters and coordinate activities among them, in such a way that only one sensor needs to be awake in each cluster to maintain network connectivity and to carry out the sensing task.

The data collected in different groups is thus likely non-redundant.

(A2) The utility of data packets originated from the same node is represented by a single utility function, in spite of the fact that they may be routed along different paths to the sinks.

(A3) The communication cost incurred after data packets arrive at the sinks is negligible. Once data packets arrives at any of the sinks, they may be relayed to other sinks via a wireline network, and hence the communication cost is assumed negligible.

We formulate the optimization problem as a convex programming problem: to maximize the total utility of data collected at sinks, subject to the channel capacity constraints and the energy constraints. For notational convenience, we define the following notions:

- $\mathcal{U}_s(x_s)$: the utility function of the data rate x_s generated from a sensor s and sent to a sink. The utility function is assumed to be a strictly concave function. The range of the source rate x_s needs to be within the range $I_s = [m_s, M_s]$;
- S_n and S_i: the set of sensors and sinks in the sensing field;
- $\mathcal{N}(i)$: the set of sources that use node i as a relay node (including node i itself);
- $\mathcal{P}(s)$: the set of nodes relaying packets for the source s (including source s itself);
- E_i: the amount of energy initially equipped with node i;
- e_i: the energy consumed in the idle state per unit time;
- e_s and e_r: the additional energy consumed in transmitting and receiving one unit of data rate per unit time;
- T_ℓ: the pre-specified, desired system lifetime.
- C_i: the channel capacity of node i;

For ease of description, in this section, we only consider the node capacity constraints in the optimization problem. The energy constrains will be considered in Section III. Similar to the problem considered by Low *et al.*[8] in wireline networks, the optimal flow control problem in wireless networks is to maximize the sum of utility of all sources subject to the node capacity constraint, C_i, at each node i. The primal optimization problem can be expressed as:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{s\in S_n} \mathcal{U}_s(x_s) \\ s.t. \quad & \sum_{s\in\mathcal{N}(i)} x_s \le C_i \qquad \forall i \in S_n \end{aligned} \tag{1}$$

Since the utility function is strictly concave, there exists a unique maximization solution for the primal problem. However, the constraints require the knowledge of all source rates. The key to making the optimization problem separable is to consider the dual problem. First we define the Lagrangian function as:

$$\begin{aligned} L(\mathbf{x},\mathbf{p}) &= \sum_{s\in S_n} \mathcal{U}_s(x_s) + \sum_{i\in S_n} p_i \cdot (C_i - \sum_{s\in\mathcal{N}(i)} x_s) \\ &= \sum_{s\in S_n} (\mathcal{U}_s(x_s) - x_s \cdot \sum_{i\in\mathcal{P}(s)} p_i) + \sum_{i\in S_n} p_i C_i, \end{aligned} \tag{2}$$

where $\mathbf{p} = \{p_i\}$ is the vector of the Lagrangian multipliers, and p_i can be regarded as the capacity price per unit bandwidth usage at node i. Next, the dual problem is defined as

$$\min_{\mathbf{p}\ge\mathbf{0}} D(\mathbf{p}) \tag{3}$$

with the objective function:

$$D(\mathbf{p}) = \max L(\mathbf{x},\mathbf{p}) = \sum_{s\in S_n} \mathcal{B}_s(p^s) + \sum_{i\in S_n} p_i C_i, \tag{4}$$

where

$$\mathcal{B}_s(p^s) = \max_{x_s\in I_s} \mathcal{U}_s(x_s) - x_s p^s, \tag{5}$$

$$p^s = \sum_{i\in\mathcal{P}(s)} p_i. \tag{6}$$

Because the second equality of the Lagrangian function in Eq. (2) can be expressed separable in x_s, the optimization of the dual problem can be solved at each node in a distributed manner. By Kuhn-Tucker theorem, the optimality point happens at: $\nabla L(\mathbf{x},\mathbf{p}) = \mathbf{0}$. Then $x_s(p)$, the unique maximizer of Eq. (5), can be obtained:

$$x_s(p) = [\mathcal{U'}_s^{-1}(p^s)]_{m_s}^{M_s}, \tag{7}$$

where $[z]_a^b \overset{\triangle}{=} \min\{\max\{z,a\},b\}$ and $\mathcal{U'}_s^{-1}(\cdot)$ is the inverse function of the derivative of utility function $\mathcal{U}_s(\cdot)$. Therefore, source s can determine its optimal rate if the information of p^s in Eq. (6), the sum of the price per unit bandwidth at each node on the path for source s, is available. On the other hand, the optimal price $\mathbf{p}$ of solving the dual problem in Eq. (3) can be obtained by applying the gradient projection method [1]. The price for node i, p_i, can be adjusted according to:

$$\begin{aligned} p_i(t+1) &= [p_i(t) - \alpha \frac{\partial D(\mathbf{p}(t))}{\partial p_i}]^+ \\ &= [p_i(t) - \alpha(C_i - x^i(\mathbf{p}(t))]^+, \end{aligned} \tag{8}$$

where $[z]^+ \triangleq \max\{z, 0\}$, α is a small positive step size and $x^i(\mathbf{p}) = \sum_{s \in \mathcal{N}(i)} x_s(\mathbf{p})$ is the sum of the rates of all flows passing through node i at time t. Finally, the solution of the optimal rate and price can be solved iteratively in Eqs. (7) and (8) at sources and routers, respectively. Low *et al.* [8] show that the optimal solution converges provided that the step size α is sufficiently small.

Price Information Dissemination*:* To determine the optimal source rate x_s, each source s needs to know the total price p^s on the path to the destination. A simple implementation method is to carry the price information of the path in the data packet, update it accumulatively from the source to the destination, and enable the destination to return the total price to the source in an explicit transport-level acknowledgment (ACK) packet. In spite of its simplicity, this method is not well-suited for resource-limited sensor networks since the overheads of explicit ACKs are considerable and reliable and timely delivery of ACK packets must be ensured.

In our proposal, the price information is conveyed through link-level ACKs in 802.11-like MAC protocols. When an intermediate relaying node receives a data packet, it retrieves the accumulative price of this flow from its flow information table [2] , and attaches the price to the link-level ACK packet. The accumulative price includes the price at the relaying node and all its downstream nodes (i.e., the nodes on the path towards the destination). Upon receipt of an ACK packet carrying the price information, the upstream node updates the price of this flow in its table. The price information of the entire path is eventually propagated from the last hop of the path to the source. By carrying the price information in the ACK packet only when the price of the flow changes, the overheads can be further reduced. A bit in the ACK packet header can be used to indicate whether the price information is included.

The price adjustment procedure in Eq. (8) executed at a node requires the sum of the rates of all the outgoing flows. This can be measured locally without incurring extra communication overheads. One drawback of the proposed approach is that each intermediate node needs to retain per-flow information. However, as sensor data are usually aggregated and processed at cluster heads, the number of data flows transported in the network is at most comparable to the number of cluster heads and hence is not prohibitively high.

[2] In the case that an intermediate relaying node always forwards packets to the same next hop towards the destination, all flows passing through it have the same path price and thus no flow table is required.

III. Linearization of Energy Constraints

How to reduce energy consumption is an important issue for battery-powered sensors and hence it is necessary to figure in energy constraints in the optimization problem. Obviously there exists a trade-off between maximizing the amount of information delivered and prolonging the system lifetime. It would be desirable to increase data rates of certain flows only when certain events of interest take place in the proximity of the sources. However, it is difficult to predict when events of interest will take place in the future. Therefore, a reasonable energy constraint is to preserve energy to ensure the system remains operational at least beyond a pre-specified system lifetime. In this section, we elaborate on how to figure in such a linear energy constraint in Eq. (1) so as to make the optimization problem separable.

Let t be the current time and $E_i(t)$ be the current remaining energy of node i. For a sensor node to operate beyond a pre-specified time instant T_ℓ, the current data rate should be determined, so that the energy incurred in the transmission, reception, and idle states for the remaining interval till time T_l is smaller than the current remaining energy $E_i(t)$. Specifically,

$$((e_s+e_r)\cdot \sum_{s\in\mathcal{N}(i)} x_s - e_r x_i + e_i)\cdot(T_l - t) \le E_i(t), \quad \forall i \in S_n. \tag{9}$$

In other word, each node needs to control the data rate (that includes traffic originating from the node and transit traffic), such that its energy consumption rate is below the rate that sustains the specified system lifetime. We call this rate the *intended energy consumption rate*, and denote it as

$$b_i(t) \triangleq E_i(t)/(T_l - t). \tag{10}$$

The optimization problem that includes both the node capacity and linear energy constraints is

$$\begin{array}{lll} \max\limits_{\mathbf{x}} & \sum\limits_{s\in S_n} \mathcal{U}_s(x_s) & \\ s.t. & \sum\limits_{s\in\mathcal{N}(i)} x_s \le C_i, & \forall i \in S_n \\ & (e_s+e_r)\cdot \sum\limits_{s\in\mathcal{N}(i)} x_s - e_r x_i + e_i \le b_i(t) & \forall i \in S_n \end{array} \tag{11}$$

Note that the energy constraint is a linear function because the *intended energy consumption rate*, $b_i(t)$ of a node i is now a parameter. Therefore, the optimization problem in Eq. (11) is a convex programming problem and can be solved in a distributed manner. Similar to the derivation in

Section II, the Lagrangian function is now defined as

$$L(\mathbf{x},\mathbf{p}) = \sum_{s\in S_n} \mathcal{U}_s(x_s) + \sum_{i\in S_n} \{p_{c_i}\cdot (C_i - \sum_{s\in\mathcal{N}(i)} x_s) + p_{l_i}\cdot [b_i(t) - ((e_s+e_r)\cdot \sum_{s\in\mathcal{N}(i)} x_s - e_r x_i + e_i)]\} = \sum_{s\in S_n} \{\mathcal{U}_s(x_s) - x_s\cdot [\sum_{i\in\mathcal{P}(s)} (p_{c_i} + p_{l_i}(e_s+e_r)) - p_{l_s}e_r]\} + \sum_{i\in S_n} (p_{c_i}C_i + p_{l_i}(b_i(t) - e_i)), \tag{12}$$

where p_{c_i} and p_{l_i} are the price charged for capacity constraint and lifetime constraint at node i, respectively. Let the accumulative prices along the path from source s to the destination for the capacity and lifetime constraints be denoted as $p_c^s \triangleq \sum_{i\in\mathcal{P}(s)} p_{c_i}$ and $p_l^s \triangleq \sum_{i\in\mathcal{P}(s)} p_{l_i}$. Similar to Eq. (7), each source s, controls its optimal rate based on the prices for the capacity and lifetime constraints:

$$x_s(p) = [{\mathcal{U}'}_s^{-1}(p_c^s + p_l^s(e_s+e_r) - p_{l_s}e_r)]_{m_s}^{M_s}. \tag{13}$$

Besides, the prices for the capacity and lifetime constraints are adapted at each node periodically based on the usage of the resource:

$$\begin{cases} p_{c_i}(t+1) = [p_{c_i}(t) - \alpha(C_i - x^i(\mathbf{p}(t))]^+, \\ p_{l_i}(t+1) = [p_{l_i}(t) - \beta[b_i(t) - ((e_s+e_r)x^i(\mathbf{p}(t)) \\ \quad -e_r x_i(\mathbf{p}(t)) + e_i)]]^+, \end{cases} \tag{14}$$

where α and β are small positive step sizes for adjusting the capacity and lifetime prices, respectively. Even though $b_i(t)$ might change with time, the price for the energy constraint is only adjusted periodically according to the gap between the current energy consumption rate and the current *intended energy consumption rate* $b_i(t)$. When the current energy consumption rate exceeds $b_i(t)$, the price for the energy constraint is increased and vice versa. Similarly, the source data rate and the price are adjusted iteratively at each node with the use of Eqs. (13) and (14), respectively.

The above approach combines both the prices for the capacity and energy constraints into a single entity. The importance of both capacity and energy constraints can be quantified in terms of the prices, based on how tight the constraints are. A more stringent constraint leads to a higher price.

IV. Estimation of Node Capacity

One major challenge of employing utility-based optimization (Eq. (1)) in the multi-hop wireless environment is how to estimate the node capacity, C_i for each node i. In contrast to the link capacity in wireline networks which is given in its link specification, the node capacity in the multi-hop wireless environment is no longer a constant but highly dependent upon the nodal distribution in its neighborhood and the traffic conditions at other neighboring nodes. In [5], a static node capacity is derived based on the conservative assumption that all the two-hop neighboring nodes are backlogged. The node capacity thus derived might be conservative when some of neighboring nodes are not backlogged.

In this paper, we take the derived node capacity, $\widehat{C}_i$, in [5] as the *nominal* capacity. In addition, we propose a lightweight capacity estimation method that adapts to the traffic conditions and buffer status. The final estimate of the node capacity is a weighted sum of the nominal capacity, $\widehat{C}_i$, and the dynamically estimated capacity, $\overline{C}_i$, i.e.,

$$C_i = \lambda\widehat{C}_i + (1-\lambda)\overline{C}_i. \tag{15}$$

The on-line estimated node capacity, $\overline{C}_i$, is calculated as the average throughput attained by all the outgoing flows, i.e.,

$$\overline{C}_i = \kappa\cdot\overline{r_i}, \tag{16}$$

where $\overline{r_i}$ is the averaged, aggregated throughput attained by all the outgoing flows, and κ ($\kappa < 1$) is a parameter that ensures system stability. As the queuing size grows indefinitely when the incoming rate is greater than or equal to the serving capacity, the parameter κ is used to prevent the system queue from growing unbounded. (We will discuss below how to on-line tune κ.) The average throughput attained by all the outgoing flows, $\overline{r_i}$, is measured over the past T_c seconds as follows. Each node records the delay, d_j, incurred by each outgoing packet P_j with packet size, $Size_j$, to the next hop, i.e. the latency from the time of retrieving the packet from the head of queue to the time of receiving the corresponding ACK packet. The average throughput attained during the past T_c seconds is calculated as

$$\overline{r_i} = \frac{1}{P}\sum_{j=1}^{P}\frac{Size_j}{d_j}, \tag{17}$$

where P is the number of packets transmitted in T_c. A low-pass filter with an exponentially weighted moving average can be used to filter out transient fluctuation of the calculated attainable throughput.

We use a simple feedback control mechanism to on-line adjust κ. The value of κ is increased by a small value, $\delta\kappa_+$, if the following two conditions hold: (i) During the past T_c seconds, no packet drops as a result of buffer overflow or failure to reach the next hop (due to contention); and (ii) the number of packets currently in the buffer is less than a pre-determined threshold, Q_{low}. On the other hand, the value of κ is decreased by a small value, $\delta\kappa_-$, if more than N_p packets are dropped due to either buffer overflow or failure to reach the next hop in the past T_c seconds. For the other cases, κ remains the same value.

V. Integrating Routing Dynamics in the Optimization Problem

The utility optimization problem considered in Section III is solved in two phases. In the first phase, the routes for all the sources are determined based on hop counts. Then in the second phase convex programming is solved given the set of the given routes. The solution based on fixed routes is not optimal, but it has been shown in [13] that the optimization problem considering both routing and flow control decisions simultaneously ($\max_{\mathcal{R}} \max_{\mathbf{x}} \sum_{s \in S_n} \mathcal{U}_s(x_s)$) is NP-hard.

Instead of searching the entire space, we propose to incorporate the optimization results into routing, and propose a modified version of Ad hoc On Demand Distance Vector (AODV) routing protocol. The proposed routing protocol provides sensors with opportunities to select routes with smaller prices and improve the overall utility of the system. The objective of integrate dynamic routing to the utility optimization problem is to increase the overall utility while keeping the system stable. Route changes should be much less frequent than rate changes in flow control in order to maintain the system stability. The trade-off between maximizing the utility and keeping the route stability is studied in [13]. The proposed routing protocol attempts to balance the trade-off between maximizing the utility and reducing routing overheads, and is composed of three phases: route initialization (*RINT*), route request (*RREQ*) and route reply (*RREP*). In what follows, we elaborate on each phase and highlight the differences between AODV and the proposed protocol. Fig. 1 summarizes the three phases.

(1) Route Initialization (RINT): In the first *RINT* phase, all the sinks construct sink-tree routes to the sensors. Unlike generic ad hoc networks, the destinations for all sensors in sensor networks are usually the sinks. In addition, sensors are usually of low mobility. Therefore, these sink-tree routes establish connections proactively between sources and destinations, and reduce the overheads incurred in route discovery.

The operations of constructing sink-trees which originate from different sinks are as follows: At the beginning of system operation, all the sinks broadcast *route initialization (RINT)* packets. A *RINT* packet carries the information of the sink ID, the number of hops traversed so far, and the accumulative energy on the path.[3] Upon receipt of a *RINT* packet, a node creates an entry in its routing table (if an entry indexed by the ID of the originating sink does not exist), storing the information of the sink ID, the ID of the node from which the *RINT* packet arrives (i.e., the next hop to the sink), the hop count, and the accumulative prices to the sink [4]. The node only forwards the *RINT* packet under the two conditions: (i) if the hop count recorded in the *RINT* packet received is less than that kept in the routing table entry (indexed by the sink ID); or (ii) if the *RINT* packet originates from a new sink and the hop count recorded in the packet is at most δh more than the hop count to the closest sink. The value[5] of δh determines the size of the overlap regions of different sink-trees. Before broadcasting a *RINT* packet, the forwarder increases the hop count in the packet by one and sets a back-off timer to reduce the possibility of colliding with *RINT* packets from other forwarder nodes. The value of back-off timer consists of a deterministic part (that is proportional to the traversed hop count h) and a uniform random part:

$$D = c_1 \cdot h + U(c_2), \tag{18}$$

where both c_1 and c_2 are system parameters. Setting of the back-off timer in this manner makes a node forward a *RINT* packet that traverses less hops earlier, and all the sink trees (originating from different sinks) grow approximately at the same rate. A *RINT* packet stops to be forwarded (and hence a sink tree stops growing) when the packet arrives in the proximity of the boundary of the "territories" of two sink nodes. A node in the overlap area has the flexibility to select any sink as its destination.

After a source chooses a sink with the least hop count as its initial destination, it determines the next-hop node as the one with the least hop count to the destination. In the case of a tie, the node with the most energy is chosen as the next-hop node. A similar forwarding rule is also applied to relay nodes. When a node receives the first packet from a new source, it relays the packet to the next hop with the least hop count to the destination. If there exist multiple next-hop candidates, the next-hop node is chosen in round robin fashion for load balancing. The per flow (source) information is also needed to be kept in intermediate routers because in the flow optimization problem the price infor-

[3] Another choice is the minimum energy of a node along the path.
[4] The initial price is the product of a default value and hop count
[5] δh = 1 in this paper to ensure routes are loop-free.

mation from downstream nodes needs to be properly propagated to the source. In addition to the flow (source) ID, the sink ID, the hop count, the next-hop ID, and the accumulative energy on the path, the routing table of a node also contains the cumulative prices for the capacity and energy constraints to the sink. Hence, at the end of the *RINT* phase, each source or intermediate node can select (based on hop count and flow count information) the next hop for the flow that originates from itself or transit flows, respectively and the flow control algorithm is executed given the routes established in the first phase.

(2) Route Request (RREQ): The routes established in the *RINT* phase may incur high prices due to the limited energy of certain intermediate node(s) or congestion along the routes. Different from route discovery in AODV, the *RREQ/RREP* phases aim to find alternate routes with smaller costs (prices) under the two conditions: (i) the overheads incurred in the *RREQ/RREP* phases are kept minimal; and (ii) in the case that a data flow f is to be switched to a new route, the "losses" of data flows that are originally routed on this new route should be well compensated by the gain in switching the data flow f to this new route.

In order not to overload the network, the restrictions on the usage of *RREQ* packets are imposed on both the sources and intermediate nodes. A source is allowed to search for an alternate route only when it detects that an event leads to high utility but the price of using the current path is so high that the data rate falls much below the possible maximal rate M_s. The original route remains operational until an alternate route with a cheaper price is found. On the other hand, selection of the next hop for either the source or relay nodes is restricted by both the hop count and the price constraints. The source sets a upper bound on the hop count to be equal to the sum of δh and the original hop count to its destination. Besides, the price of the new route must be cheaper than the original price by a certain level; otherwise, it may not be worthwhile to distribute the *RREQ* packet. (The details on how to determine whether or not the price along an alternate path is sufficiently cheap will be given below.) Furthermore, to prevent excessive exchanges of *RREQ* and *RREP* packets, *RREQ* packets are sent by unicast rather than broadcast. The node with the cheapest price in the routing table is chosen as the candidate of the next hop. A *RREQ* packet contains the source ID, the original sink ID, the upper bound on the hop count, the original price, the value of the event detected at the source, and finally the accumulative prices, $p_{c_{up}}$ and $p_{l_{up}}$, for both the capacity and lifetime constraints of all upstream nodes (including the source) on the new route.

Recall that the purpose of changing the route is to increase the *overall* utility of the system. As such, we need to consider not only the utility gain of the flow that is be switched to a new route, but also the utility loss of all the flows that are originally routed on the new route (due to the increased traffic from the redirected flow). The criterion for accepting a route change is as follows: Suppose a source s changes its route from path $\mathcal{P}_o(s)$ to path $\mathcal{P}_n(s)$. The route change leads to the utility changes of all the flows using node s or nodes on $\mathcal{P}_o(s)$ or $\mathcal{P}_n(s)$. In order to increase the system utility, the change in the utility of the overall system

$$\sum_{i:\mathcal{P}_o(i)\cap\{s,\mathcal{P}_o(s),\mathcal{P}_n(s)\}\neq\emptyset} (\mathcal{U}_i(x_{i_n}) - \mathcal{U}_i(x_{i_o})) \qquad (19)$$

should be positive, where x_{i_n} and x_{i_o} represent the new and old rate of node i. The flows using $\mathcal{P}_o(s)$ benefit from the reduced traffic and thus

$$\sum_{i:\mathcal{P}_o(i)\cap\{\mathcal{P}_o(s)\}\neq\emptyset} (\mathcal{U}_i(x_{i_n}) - \mathcal{U}_i(x_{i_o})) \geq 0. \qquad (20)$$

As a result, the following inequality holds:

$$Eq.\ (19) \geq \sum_{i:\mathcal{P}_o(i)\cap\{s,\mathcal{P}_n(s)\}\neq\emptyset} (\mathcal{U}_i(x_{i_n}) - \mathcal{U}_i(x_{i_o})). \qquad (21)$$

Therefore, as long as the utility change of all the flows using node s or nodes on $\mathcal{P}_n(s)$ is greater than zero, the change in the utility of the overall system is positive. Testing of the utility change is performed at all the nodes that receive RREQ packets. The challenge is how to estimate the new rate x_{i_n} after the route change. The approximation we use is that the intermediate forwarder node f determines its own price change and the new rate of all flows passing through itself in the following six steps:[6]

(S1) Node f retrieves the cheapest price towards any sink satisfying the hop count constraint. Since this price is calculated without consideration of the redirected traffic, we denote it as the *priori* price, p^f_{pri}.

(S2) Node f estimates the data rate of source s based on the value of the event (carried in the *RREQ* packet) and the sum of the upstream price and the priori price[7]:

$$x_{s_{pri}} = [\mathcal{U'}_s^{-1}(p_{up} + p^f_{pri})]^{M_s}_{m_s}. \qquad (22)$$

[6] For simplicity, only the capacity price is presented here.

[7] we assume the forwarder knows the function of $\mathcal{U}_s(\cdot)$ as long as the value of the source is known.

(S3) According to the increased traffic from the redirected flow, node f calculates its $posterori$ price, p^f_{post}, by Eq. (14).

(S4) Similar to (S2), the *posteriori* data rate of source s is determined based on the price $p_{up} + p^f_{post}$.

(S5) Similar to (S1)-(S4), for each flow from node i passing through node f, node f first determines the original price of node i. Next the price increase of node f is added to node i's price. Finally the new rate is estimated based on the new price.

(S6) Node f calculates the utility change given in Eq. (21) and determines whether to forward the *RREQ* packet based on the following criterion:

$$\mathcal{U}_s(x_{s_n}) - \mathcal{U}_s(x_{s_o}) \geq c_3 \cdot [\sum_{i \in \mathcal{N}(f)} (\mathcal{U}_i(x_{i_o}) - \mathcal{U}_i(x_{i_n}))]. \tag{23}$$

Only when the gain in the utility increase of source s (the term on the left hand side in Eq. (23)) is greater than or equal to c_3 ($c_3 > 1$) times of the utility decrease in all existing flows, the RREQ packet is forwarded.

By applying a price estimation method, we can reduce the impact of redirecting a data flow on the flows that are originally routed on the new route. Finally, the forwarding process continues until the *RREQ* packet reaches to the last hop on the new path, at which point the *RREP* phase commences.

(3) Route Reply (RREP): When a *RREQ* packet reaches the last hop, the node next to the sink sends back a *RREP* packet with the price of the new route equal to its own price[8]. Similar to the price estimation method (S1)-(S4) in the *RREQ* phase, the posterior price is calculated by figuring in the traffic increase due to the redirected flow. Furthermore, the sum of the utility loss of all flows passing through each node on the new route is calculated (following (S5)-(S6)). Both the posterior downstream price and the utility loss are carried in the *RREP* packet. All the intermediate relay nodes follow the same procedures to calculate the downstream price and accumulative utility loss. When the source receives the *RREP* packet, it determines whether to change its route based on Eq. (23).

Phase I: Route Initialization

Sink-tree construction:
1. Sinks: broadcast a *RINT* packet
2. Forwarders: upon receipt of a *RINT* packet:
3. if (RINT packet satisfies the hop count constraint)
4. hop count++
5. set a back-off timer based on the hop count
6. broadcast the *RINT* packet

Selection of the initial route:
1. Sources: choose the destination and the next hop based on the hop count information
2. Forwarders: choose the next hop based on the hop count and flow count information

Phase II: Route Request

Sources:
1. if (an event with high value $\&\&$ too expensive price)
2. unicast *RREQ* to the node with the cheapest price

Forwarder f: upon receipt of a *RREQ* packet:
3. follow (S1)-(S6) to calculate the posterior price and utility change
4. if (utility change $< 0 \,||\, f$ is the last hop)
5. sends back *RREP*
6. else
7. forwards *RREQ* to the node with the cheapest price

Phase II: Route Reply

Forwarders: upon receipt of a *RREP* packet:
1. follow (S1)-(S6) to calculate the posterior price and utility change and append both to the RREP

Source: upon receipt of a *RREP* packet:
2. if (cheaper price $\&\&$ utility change > 0)
3. redirects the flow to the new route
4. else
5. stays in the original route
6. sends another *RREQ* if a potential good route exists

Fig. 1. Three phases of the proposed dynamic routing algorithm.

VI. SIMULATION RESULTS

We evaluate the performance of the distributed utility-based approaches using the *ns-2* [16] simulator. A total of 60 sensors and 4 sinks are deployed in a square grid given in Fig. 2. The distance between neighboring sensors is 200 meters. We assume the radio transmission range is 250 meters. Therefore, each sensor has at most 4 neighbors. The data transmission rate of the wireless channel is assumed to be 40 kbps. The utility function used in the simulation is defined as $\mathcal{U}_s(x_s) = v_s \cdot log(x_s + 1)$, where v_s and x_s are the utility value of a packet and the source rate (in units of #packets/second) for sensor node s, respectively. The function $\mathcal{U}_s$ is a non-decreasing and concave function of node s's sending rate. To test the performance under dynamic environments, the values of packets are not constant, but are specified according to Figure 3 and Table I. We envision a volcano monitoring system used to record the volcano activities. When a volcano stays at the dormant status (state 1), sensors transmits data at a low rate to

[8] We assume the sinks have unlimited resource in both outgoing capacity and energy resource and hence the price of sinks is equal to zero.

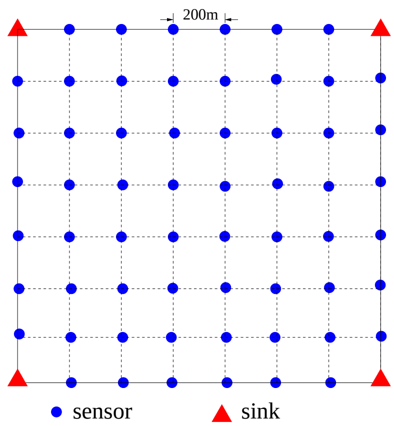

Fig. 2. The topology used in the simulation study. A total of 4 sinks and 60 sensors are deployed on a square grid. The distance between neighboring sensors is 200 meters, and each sensor has at most 4 neighbors.

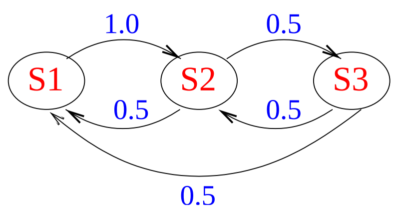

Fig. 3. Markov state model of value of packets at a sensor.

sinks. If sensors detect abnormal events (state 2), a higher data rate is required to facilitate transport of data back to the sink for further analysis. Once a volcano eruption event is likely to take place (state 3), the transmission rate should be further raised.

We use a Markov chain model given in Fig 3 to generate the value of packets captured at each sensor. The parameters used in each of the Markov states are listed in Table I. The time period of a state is uniformly random distributed in [10, Max. Period] seconds. The parameters for power consumption follows the setting in [4], i.e., the power consumption incurred in the transmission, reception, and idle state is 1.4, 1.0, and 0.83 W, respectively. Hence, e_i is 0.83 W. e_s and e_r are the additional energy consumed (in addition to e_i) in sending and receiving a bit of data, and are equal to the product of T_b, the time to send a bit and

State	Max. Period	Value	$M_s(bps)$	$m_s(bps)$
S1	200 seconds	1	100	25
S2	50 seconds	10	500	25
S3	100 seconds.	100	3000	25

TABLE I
PARAMETERS OF THE MARKOV STATE MODEL.

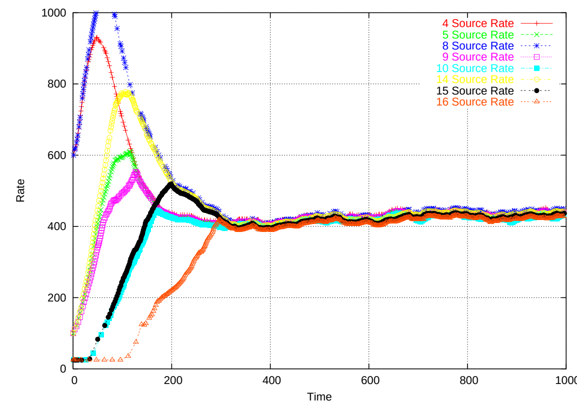

Fig. 4. Data rates of the eight sensors in the upper left quarter of the 36-node square grid.

0.57(=1.4-0.83) and 0.17(=1.0-0.83), respectively. (Note that the units of e_s and e_r are joules per bit.) The payload size of a packet is set to be 70 bytes (including 20 bytes of IP header but not MAC and PHY headers).

System stability in the case that node capacities are on-line measured*:* Before evaluating the utility-based approach in dynamic environments, we first verify whether or not the data rates of sources converge under the case when the node capacites are on-line estimated. In the first set of simulations, a total of 32 sensors and 4 sinks are deployed in a square grid in the same manner as Fig. 2, except that each row or column has 6 nodes. (The four sinks located at the corners are labeled as 0, 1, 2, and 3. The 32 sensors are labeled as 4, 5, . . ., 35 in sequence, from the top row to the bottom row, and from left to right.) The value of the data from all sensors is fixed to be one, all sensors are initially equipped with 1000 joules (E_i), and the node capacity of each node is on-line measured and calculated. Fig. 4 gives the results of data rates from eight sensors located in one quarter of the square grid. At the beginning of the simulation (0-100 seconds), only sensor nodes 4 and 8 — the two sensors next to the sink – have high data rates because they do not incur prices from the other sensors. All other sensors have low rates due to their high default prices. After the time instant 300 seconds, the data rates of all the sensors become stable at approximately 400 bps till the end of simulation.

Advantages of on-line capacity estimation*:* In the second set of simulations, we compare the performance of the proposed utility-based approach with and without on-line nodal capacity estimation enabled. The sensor network given in Fig. 2 is used. The value of the data changes according to the Markov model given in Fig. 3. All the 60 sensors are initially equipped with 1000 joules. Figure 5

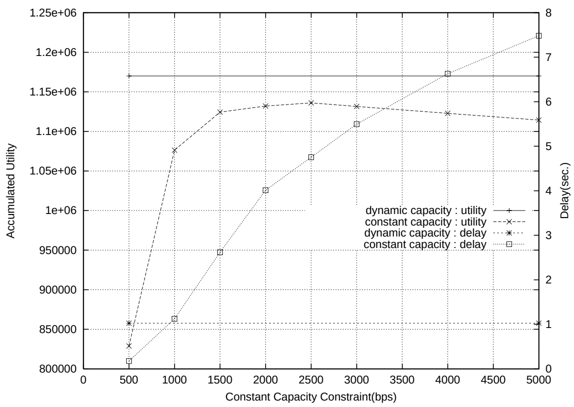

Fig. 5. The performance of the proposed utility-based approach, when the node capacity is on-line estimated and when it is fixed with 10 different values (labeled in the x-axis).

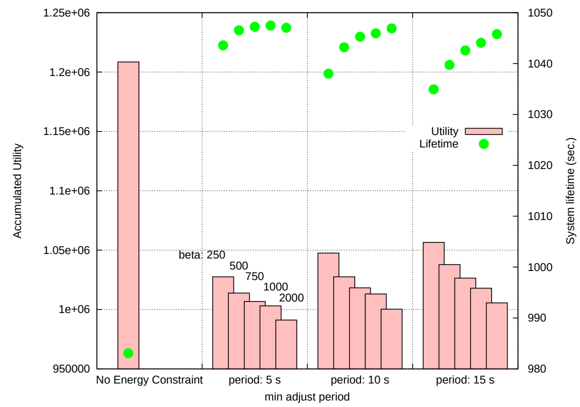

Fig. 6. The utility and system lifetime for the proposed approaches with and without the lifetime price. Different values of the step size, β, and the adjustment period are considered.

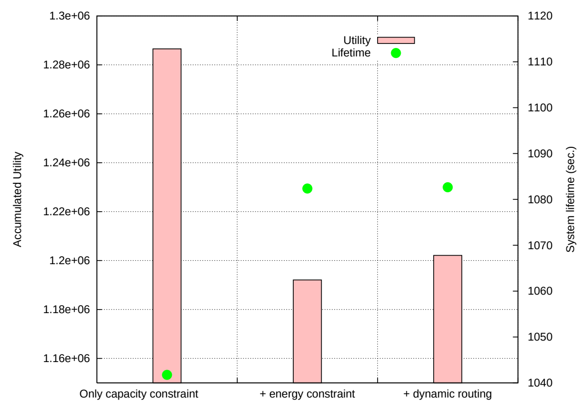

Fig. 7. Performance comparison of three different versions of the proposed utility-based approaches (from left to right): (i) consideration of only the capacity constraint, (ii) consideration of both the capacity and energy constraints, and (iii) consideration of both the two constraints and dynamic routing.

gives the result of the proposed approach, when the node capacity is on-line estimated and when it is fixed with 10 different values. The average delay of the proposed approach with fixed capacity values increases as the capacity values increase, while the utility is maximized at the capacity value of approximately 1400 bps. The utility achieved by the proposed approach with on-line estimated node capacities is higher, while the end-to-end latency incurred is smaller. This demonstrates the advantages of on-line estimating the node capacity.

Effects of adjusting lifetime prices*:* The effect of how to adjust the lifetime price (Section III) on the utility and the system lifetime is investigated in the next set of simulations. Initially each sensor is equipped with the same energy, 1000 joules. The intended system lifetime is set to 1050 seconds. (Note that the maximum achievable system lifetime is approximately $1000/0.83 \approx 1200$ seconds, where 0.83 watt is the power consumed in the idle state.) Fig. 6 gives the performance comparison of approaches with and without lifetime price adjustment. In particular, we vary two parameters in the approach: the step size used to adjust the lifetime price, β (Eq. (14)) and the minimum adjustment period. As expected, the proposed utility-based approach without the energy constraints achieves the highest utility, but the system lifetime is also much shorter. The trade-off between the utility and the system lifetime is also observed in the selection of the step size used to adjust the lifetime price, β (Eq. (14)) and the minimum adjustment period. Larger values of β or shorter adjustment periods increase the impact of the lifetime price on the system performance and thus lead to longer system lifetime but less utility.

Results of dynamic routing*:* In this set of simulations, we evaluate the performance of the proposed approach with dynamic routing proposed in Section V. One difference in the set of simulations is that instead of having the same amount of battery capacity, each node is equipped with a battery capacity uniformly distributed in the range of [1000, 2000] joules. Setting different battery capacities for different nodes better simulates deployment of sensors in realistic environments. The intended system lifetime is set to 1200 seconds. Fig. 7 gives the results of three different versions of the proposed approaches: (i) consideration of only the capacity constraint, (ii) consideration of both the capacity and energy constraints, and (iii) consideration of both the two constraints and dynamic routing. As compared with the approach with static routing, the approach with dynamic routing performs better both in terms of the utility and the lifetime, although the performance improvement is not dramatic. This is perhaps due to the fact that the energy consumed in overhearing is not considered in the energy constraint of the formulated problem. Although the approach with dynamic routing selects an alternate route to avoid use of nodes with low battery levels, the new route may still be within the carrier sense range of the nodes with low battery levels. In this case, the system lifetime can not be significantly increased.

VII. Conclusions and Future Work

In this paper we propose a distributed, energy-aware, utility based approach to improve the system utility of wireless sensor networks, subject to both the channel capacity and energy constraints. Our major contributions are (i) on-line estimation of node capacity (to be used in the capacity constraint of the optimization problem) in the multi-hop wireless environment, (ii) inclusion (and linearization) of energy constraints that relate the system lifetime to the data rate, and (iii) incorporation of optimization results in selecting routes that maximize the utility. The simulation results indicate that the utility-based approaches balance between system utility and system lifetime, and can serve as an effective resource utilization mechanism for different types of wireless sensor networks.

As mentioned in Section VI, the performance of the proposed approach with dynamic routing is not dramatically improved, as the the energy consumed in overhearing is not considered in the energy constraint of the formulated problem. As part of our future work, we plan to incorporate the energy consumed in overhearing into the problem formulation. Alternatively, we may keep the problem formulation unchanged, but instead incorporate power-off operations into the proposed utility-based approach. That is, we turn off the radio circuitry of nodes with low battery levels, if they do not forward packets for other nodes (as determined by the proposed approach). These nodes are then periodically turned on to check if they are on the new routes of certain redirected flows. The interaction between the utility loss due to the latency caused by power-saving operations and the gains in the increased system lifetime is an interesting research issue under investigation.

References

[1] D. P. Bertsekas, "Nonlinear Programming, " 2nd edition, Athena Scientific, Belmont, MA, 1999.

[2] J. Byers and G. Nasser, "Utility-Based Decision-Making in Wireless Sensor Networks (Extended Abstract)," Proceedings of IEEE MobiHOC 2000, Boston, MA, August 2000, pp. 143-4.

[3] J-H Chang , L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," in INFOCOM 2000.

[4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," in ACM Wireless Networks Journal, Volume 8, Number 5, September, 2002.

[5] W-P Chen and L. Sha, "An Energy-Aware Data-Centric Generic Utility Based Approach in Wireless Sensor Networks," Proc. of the Third International Symposium on Information Processing in Sensor Networks (IPSN), April 2004.

[6] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *In Proc of the Sixth Annual IEEE/ACM International Conference on Mobile Computing and Networking*, August 2000.

[7] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," Journal of Operational Research Society, vol. 49, no. 3, pp. 237V252, March 1998.

[8] S. H. Low and D. E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," IEEE/ACM Transactions on Networking, 7(6):861-75, December 1999.

[9] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing," Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.

[10] Y. Qiu and P. Marbach, "Bandwidth Allocation in Wireless Ad Hoc Networks: A Price-Based Approach," in IEEE INFOCOM, San Francisco, April 2003.

[11] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Pricing and Power Control in a Multicell Wireless Data Network," IEEE Journal on Selected Areas in Communications, Vol. 19, No. 10, pp. 1883-1892, Oct. 2001.

[12] N. Sadagopan and B. Krishnamachari, "Maximizing Data Extraction in Energy-Limited Sensor Networks," in IEEE INFOCOM, Hong Kong, March 2004.

[13] J. Wang, L. Li, S. H. Low and J. C. Doyle, "Can TCP and shortest path routing maximize utility," Proceedings of IEEE INFOCOM, San Francisco, April 2003

[14] Y. Xu, J. Heidemann, D. Estrin, "Geography-informed Energy Conservation for Ad-hoc Routing," *In Proc. of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (ACM MobiCom)*, Rome, Italy, July 16-21, 2001.

[15] Y. Xue, B. Li, K. Nahrstedt, "Price-based Resource Allocation in Wireless Ad Hoc Networks," in the Proceedings of the Eleventh International Workshop on Quality of Service (IWQoS 2003)

[16] UCB, LBNL, "VINT network simulator," http://www-mash.cs.berkeley.edu/ns/.