JOINT OPTIMIZATION OF A PROTOCOL STACK FOR SENSOR NETWORKS

R. C. Shah, A. Bonivento, D. Petrovic, E. Lin, J. van Greunen and J. Rabaey
University of California, Berkeley
Berkeley, CA

ABSTRACT

In this work, we present protocols for routing, MAC and power-control and jointly optimize these protocols. The joint optimization aims to capture the impact of cross-layer interaction on the efficiency and performance of the protocol stack. We model the protocol stack as a whole, rather than a collection of individually modeled layers. In our model, the protocol stack has tunable parameters that affect the operation of each layer to achieve globally specified performance and efficiency. In the paper, we begin by presenting the randomized protocols at each layer that exploit node density in order to achieve reliable communication. First, we present a region-based opportunistic routing protocol. Then, at the medium access layer, we consider an asynchronous rendezvous scheme called TICER.. Third, a randomized sleep discipline is set forth that allows nodes to power down periodically. Finally, we combine the routing, MAC and power-control protocols to obtain the constrained optimization problem. Results show that it is possible to minimize energy consumption while satisfying application requirements on end-to-end delay.

INTRODUCTION

The primary goal of this paper is to present a holistic view of the sensor network protocol stack. In existing sensor network research, the protocol stack is divided into layers and each layer is optimized separately. While this approach leads to good performance of individual layers, it fails to account for the interaction between layers and the impact that design choices in one layer have on the rest of the protocol stack. We believe that additional performance gains can be obtained by optimizing the layers jointly to take advantage of cross-layer interaction. Thus, we present a unified protocol stack in this work.

This unified protocol stack must be optimized to reduce power consumption, be able to meet certain application requirements and be robust to faults and failures in the network. To fulfill these requirements, this work follows a two-pronged approach. First, robust algorithms were designed for each layer in the protocol stack. These algorithms utilize randomness to ensure against faults and to improve performance. The use of randomness also makes the algorithms compatible with each other so that they can easily be optimized together. The second prong of the approach was to put the algorithms together and jointly optimize their parameters to make sure performance requirements are met as efficiently as possible. The solution to the resulting optimization problem provides the optimum sleeping time for the TICER as well as the sleep discipline algorithm, and the optimal next-hop size for the routing protocol. This joint optimization is a starting point for combining protocols and obtaining optimum parameter values for many different algorithms at once.

As mentioned above, randomness is an integral part of our approach. Randomness brings redundancy to the protocol stack, which increases robustness, and allows nodes to only use resources as they become available. As an illustration of the principle of exploiting randomness, in [1], it is shown that it is optimal to transmit information between two wireless antennae only when conditions are favorable. This type of transmission is called opportunistic, because the transmitter transmits opportunely when the channel is strong.

In the context of sensor networks, we utilize the random nature of the channel to make the use of efficient use of the limited energy supply at nodes. Our basic philosophy is to utilize the randomness of the network environment to obtain better performance rather than trying to fight and nullify it. By network environment, we refer to not only the channel quality, but also nodes that may run out of energy, move around the network, or in general have a random component to their availability.

Including randomness in the design philosophy permeates the PicoRadio protocol stack. Starting at the network layer, regions in space that are useful for forwarding packets are selected, rather than a single next hop node that may turn out to be unavailable. These forwarding regions would consist of a number of nodes, and it is likely that at least one of them may be available for packet forwarding and have a good channel with the current node. The MAC layer is also built on the presumption of channel quality changes, hence does not try to maintain neighbor lists or keep statistics for each neighbor. When a packet needs to be transmitted, the channel quality is measured, using the forwarding region information that is passed down to the MAC layer from the network layer. Based on the channel quality at that point in time, the MAC layer instantaneously decides on the best node to forward the packet. This entire mechanism ensures that the protocols do not try to impose a graph-like structure on the network, preferring to embrace the randomness of the channel.

The paper is organized as follows: Opportunistic routing is introduced in the next section, followed by a discussion of the MAC layer protocol called TICER. Then the sleep discipline (duty-cycling of nodes) is presented. Duty cycling affects the availability of nodes for routing and is thus an integral part of the protocol stack. After the protocols are introduced we set forth the joint optimization model. Finally, the paper concludes.

OPPORTUNISTIC ROUTING

This section describes region-based opportunistic routing. Opportunistic routing is based on geographic routing, but instead of choosing a single node as the next hop, a set of "equivalent" nodes is chosen. The intuition behind this algorithm is that if there is sufficient density of nodes, there will be at least one next-hop node to which the channel is good. This set could be specified in many ways – an explicit list of nodes, a region in space or some other mechanism. Since our concept is rooted in geographic routing, we choose a region in space as the forwarding region. The shape of the region is can be varied for different routing results. Specifically, as shown in [3], a lens formed by the radio range of the current node and the circle centered at the destination is the optimal forwarding shape. We can also generalize the concept of a lens when the radio range is not circular to the region that is more than a certain distance away from the current node.

The reason for choosing a geographic routing scheme is because of its efficiency in dense networks. The biggest advantage of geographic routing is the very low route setup information needed and also the fact that the routing tables are very small. As shown in [2], the size of the routing tables is O(log N) where N is the number of nodes in the network. In addition, all nodes in the network know their position due to the application requirements, hence geographic routing does not impose any extra overhead of a locationing algorithm.

Now, once the forwarding region is specified, the challenge is in identifying the nodes that currently have a good channel. Since this changes frequently, this check has to be done at the time of packet transmission. Also, the time scale involved for the network layer to obtain this information and adapt to it is much longer than the channel coherence time (channel coherence time is the expected amount of time for which the channel gain remains constant). This means that the final decision of the node to use for packet forwarding needs to be done at the MAC layer. When specifying the forwarding region, the network layer can also provide sufficient *hints* to the MAC layer about the choice of next hop, which can then probe the channel conditions and based on both these factors, select a good next hop node.

Hence at a very basic level, opportunistic routing provides a framework for redefining the traditional services provided by the protocol layers to better suit the wireless channel and low energy/low cost sensor nodes. This framework defines the functionality expected of the network and MAC layers, with many possible implementations of each of them. At the network layer, the routing protocol needs to specify a set of nodes (as opposed to a single node), which can be used for routing. At the MAC layer, the protocol should accept a set of forwarding nodes from the network layer. From this it has to select a single node for forwarding based on the network layer directions and send the packet to that node. Many different MAC implementations are possible based on the signaling between nodes, and how the forwarding nodes are chosen based on the network layer's hints.

Moreover, in order to achieve high power efficiency, one straightforward way is to power off sensor nodes as often and as long as possible, given the characteristics of typical sensor networks—relatively rare communication (0.1 to 10 packets/second) and short packets (less than 500 bits). With this approach, however, communication between any two nodes is possible only if both of them are powered on simultaneously. Therefore, it is necessary to arrange simultaneous on-time for nodes wishing to communicate, a method referred to as a *rendezvous scheme.*

MAC LAYER IMPLEMENTATION

In this section we introduce one particular rendezvous scheme, namely, Transmitter Initiated CyclEd Receiver (TICER) [4]. This scheme belongs to the Pseudo-Asynchronous category of rendezvous schemes, in that nodes establish rendezvous on demand (pseudo-asynchronous), while using an underlying periodic wakeup scheme (cycled receiver).

For the sake of simplicity, we first focus on a single destination node in the following discussion. When a sensor node has no data packet to transmit, it wakes up to monitor the channel every T seconds and goes back to sleep after a wakeup duration T_{on}, as shown in Fig. 1. T is selected according to the sleeping discipline explained in the next section. When the node has a data packet to transmit ? either generated from the upper layers of the protocol stack or forwarded by another node? it wakes up and monitors the channel for duration of T_{on}. If it does not hear any ongoing transmissions on the channel, it starts transmitting request-to-send (RTS) signals to the destination node, and monitors the channel for a time T_l for responses after each RTS transmission. The destination node, upon waking up according to its regular wakeup schedule, immediately acquires and receives the RTS's, and responds with a clear-to-send (CTS) signal to the source node. Upon receiving the CTS signal, the source node starts to transmit the data packet. After correctly receiving the data packet, the destination node ends the session with an acknowledgement (ACK) signal transmitted to the source node. This protocol is illustrated below in Figure 1.

The issue becomes somewhat more complicated when TICER is used along with opportunistic routing. Here multiple nodes are targeted by an RTS, thus there is a potential problem of collision if all the nodes in the forwarding region reply immediately with CTS packets. The way we handle this is that when any node in the forwarding region receives an RTS, it chooses a random backoff time. When a node is ready to transmit the CTS packet, it senses the channel for a time T_{sense} before transmitting. If the channel is idle, it goes ahead with the CTS transmission, else it goes back to sleep. In this way, we avoid collision of the CTS packets in most cases. Subsequently, the protocol is similar to the version of TICER described above – the sending node transmits the data packet which is then followed by an ACK signal.

Some rules that should be considered when designing the TICER protocol:

- Since longer wakeup duration T_{on} implies more monitor power with no gain in performance, T_{on} should be kept as short as possible. However, T_{on} has to be long enough for a waken-up node to completely receive one RTS. Therefore, it should be long enough to receive at least two RTS beacons. In this way, a node will neither wake up in between two RTS's and miss rendezvous, nor receive partial RTS packets.
- To minimize transmit power, the control packets RTS, CTS and ACK (with length T_b) are made as short as possible. These packets contain the local MAC addresses of the source and destination node(s), and a preamble for acquisition.
- To reduce collision rate, we introduce an extra control channel. All control signals, including RTS, CTS, and ACK, are sent in the control channel, and only the data packets are sent in the data channel. This however, requires support from the physical layer as well.

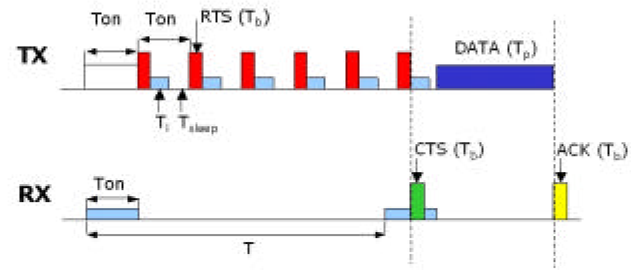

Figure 1. TICER Scheme

One variable, which impacts the MAC performance at a node, is the time T a node sleeps before it wakes up and participates in packet forwarding. A larger value of T leads to power savings due to a low node duty cycle, however, that leads to more RTS beacons before a node can rendezvous with its target node. In addition the latency also increases. Thus the sleep discipline has a strong impact on the system power performance and needs to be optimized carefully.

SLEEP DISCIPLINE

The last protocol that we present in this paper is the duty-cycling algorithm or sleeping discipline. In order to concisely discuss a sleeping discipline that is compatible with the rest of the protocol stack (routing and MAC layer), the scope of the sleeping discipline presented in the paper has been narrowed to the application of delay constrained opportunistic routing. In other words, it is assumed that the nodes use opportunistic routing, and the performance constraint is given by a delay specification. The per-hop delay constraint is specified by a tuple $\{t, p_t\}$ and is met when, with probability p_t, the hop-delay will be less than t, *i.e.*, $P(\text{Per-hop delay} > t) < p_t$. Note: the per-hop delay constraint can be translated into an end-to-end delay constraint if the number of hops and joint distribution of the per-hop delays are known.

The proposed sleeping discipline is extremely lightweight, requires no communication overhead between nodes, and is robust to node failures. The algorithm enables individual nodes to adapt their duty cycle to meet global performance requirements for the sensor network. The amount of work an individual node must perform to meet these global performance requirements is usually proportional to factors such as traffic patterns and node density. These factors are not known a priori and may vary over time. Significant savings in energy dissipation can result from an adaptive scheme.

The following assumptions are made:

- A node can process and store all packets that are forwarded to it. In other words, the amount of traffic is low enough that it never exceeds a node's buffer space, or nodes have infinite buffering capacity.

- When a packet is transmitted, the transmitting node appends a timestamp to the packet, specifying how long it has been waiting to transmit the packet.

The main idea behind the sleeping discipline is that the performance (delay) constraint will be met when the combined node wake-up activity in a region is above a certain level. In order to ensure that individual node participation, which is distributed, is high enough, each node needs to know the overall activity in its region. If the packet arrivals are memory-less, the node can estimate how long it has been since the last awakening of a node in its region, without communicating with another node. This estimate is made based on how many packets are waiting to be forwarded when the node wakes up, and how long they have been waiting. The packet arrivals are determined by the wake up events of the previous hop nodes. Therefore, having a memory-less node wake-up process ensures that packet arrivals also form a memory-less process.

Using an exponential wake-up scheme ensures that the node wake-ups are memory-less. In addition, the per-hop delay is the amount of time a packet must wait until the first node in a region wakes up. Under the exponential wake-up scheme, this time can be statistically modeled as the minimum of N exponential variables, where N is the number of nodes in the region. More concisely, the per-hop delay is exponential with parameter m (m is the combined wake-up frequency of the equivalent nodes). The per-hop delay depends only on the total (sum) wake-up rate of equivalent nodes and not an individual node's wake-up rate. Thus, there is an optimal aggregate wake-up rate m^* that minimizes the node wake-up rate while satisfying the delay constraint.

The intuition behind the algorithm is that nodes can pre-compute the optimal wake-up rate, m^*. Every time a node wakes up, it receives packets that are waiting to be processed by the set of equivalent nodes i belongs to. Based on the arrivals of packets, the node forms an estimate of the aggregate wake up rate of its equivalent nodes, and decides to wake up more/less often depending on whether the estimated wake up rate is less/greater than m^*.

There are two crucial parts to the algorithm. First, the estimate of the aggregate wake-up rate for a region is essential. Second, load sharing (fairness) between nodes is important. In order to achieve fairness, the algorithm must ensure that the nodes' independent adjustments converge to a fair equilibrium where the load is equally distributed between equivalent nodes. For brevity, the details of these two aspects (aggregate wake-up rate estimation and fairness) are omitted from this paper. The sleep discipline, including the aforementioned aspects and results, is presented in more detail in [5].

The discussion of the sleep discipline concludes with the following step-by-step description of the algorithm. The algorithm is executed independently at every node as soon as the node wakes up. For a node i located in a region B the algorithm consists of the following steps:

1. Wake up and receive all k waiting packets, over a short, fixed period. Note that node i does not accept additional packets after this period.
2. Use the longest time a packet has been waiting, to estimate the time since the last wake-up in region B.
3. If the aggregate wake-up rate for the region is less than the optimal, m^*, node i multiplicatively decreases its average sleep time. Conversely, if the estimated wake up rate in the region is greater than m^*, node i additively increases its average sleep time.
4. Stay awake and beacon until all k packets have been forwarded (exhaustive service discipline).
5. Draw a value x from an exponential distribution with mean $1/m_i$ and go to sleep for the duration of x. When node i awakens return to step 1.

The next section presents the joint optimization of the three algorithms discussed above.

PROTOCOL OPTIMIZATION

In this section we introduce the cross-layer optimization for dense sensor networks. We show how to optimize the opportunistic routing, MAC and the exponential sleeping discipline jointly. The goal of the joint optimization is to minimize the total energy consumption while satisfying performance requirements. As mentioned above, the performance requirement considered in this paper is the end-to-end delay constraint for routing the packets.

In order to formulate the problem as an optimization problem, a cost function and a constraint function need to be formulated. For this paper, the cost function is the energy consumed when routing packets, and the constraint function is the end-to-end probabilistic delay of the packets. To quantify the power consumed when routing packets, it is essential to have a model of the network topology, and of the physical layer. Both these aspects affect the power consumption of the sensor nodes. The network topology and routing model is introduced in the following section, while parameters from the physical layer will be defined as they are used in the cost and constraint functions. After the network model has been discussed, the cost function and constraint function will be presented.

NETWORK MODEL

In this paper a simplified network model is used. The network has a single controller. All packets in the network are routed toward the controller. The network topology is laid out along a single-dimension as illustrated in Figure 2. For the purpose of opportunistic routing, nodes are divided into rectangular blocks (regions) and packets are forwarded from a node in one block to any node in the adjacent block. There are h blocks in the network, the i^{th} block has size d_i, and the distance from the controller to the outermost block is D. Also, the wake-up rate of nodes (which is a parameter of the sleeping discipline) in block i is denoted m_i. In [6] these optimizations are considered for more general network topologies.

The channel between any pair of nodes is modeled as a Bernoulli random variable. Each time a node sends a beacon signal to a neighboring node the beacon is received with probability p. However, if the beacon was successfully received we assume that the packet exchanges following immediately afterwards are within the coherence time of the channel and are thus also successfully received. As a result nodes wake up on average $\frac{1}{p}$ times more often in order to ensure the same performances they would have with a non-fading channel.

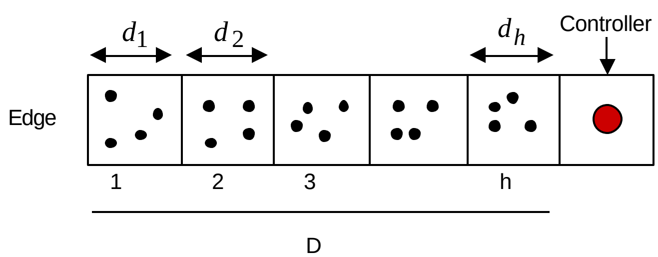

Figure 2: Network model with single sink (controller) and routing blocks

OPTIMIZATION COST FUNCTION

To derive the cost function we need to characterize the physical properties of the nodes. Then the energy consumption required by an individual node in the routing process is quantified. Finally, the energy consumption of individual nodes is added to obtain the energy cost for routing a packet through the entire network. First, two sources of node energy consumption are considered: transmissions/receptions and node-wake-ups.

1. *Transmission & Reception*: when a packet is transmitted, the distance the packet is sent determines the required transmission energy. This is a dynamic cost because it depends on the routing hop-size. In this paper it is denoted as E_{Txdyn}. This dynamic cost can be written as $E_{Txdyn} = rd^b$, where *?* is the required transmitted energy for a receiver 1 meter away (*?* can be measured given a radio), and *ß* is the roll-off factor as the signal decays with distance. With this formulation, as the hop-distance d decreases, the dynamic cost decreases. In practice there is also a fixed energy cost per hop to receive a packet. This fixed cost is denoted E_{RF}. Also, with the TICER MAC protocol, nodes beacon when they have a packet to send. The beacon transmission energy also depends on the hop-distance and is denoted E_{Wdyn}.
2. *Node wake ups*: nodes consume energy when they monitor the channel for packets. This is essentially "idle" mode energy consumption, and is denoted E_{Id}.

In this paper it is assumed that all the packets are generated at the boundaries of the network at rate λ. Note: if the end-to-end delay constraint is met for packets routed from the boundary of the network, it should also be met for packets routed from the middle of the network over a shorter distance.

Combining the energy required per node, the activity of nodes in the network and the rate at which packets are generated and forwarded, the total energy cost of the network over a period T can be obtained. This total cost is shown below in (1). A more detailed derivation is presented in [6].

$$E_{tot} = T\left(l\left(hR + \sum_{i=1}^{h} E_{TxDyn\_i} \right) + \frac{1}{p} \sum_{i=1}^{h} m_i \left(E_{Id} + E_{Wdyn\_i} \right) \right) = T\left(l\left(hR + \sum_{i=1}^{h} r d_i^{b} \right) + \frac{1}{p} \sum_{i=1}^{h} m_i \left(E_{Id} + g d_i^{b} \right) \right) \quad (1)$$

OPTIMIZATION CONSTRAINT FUNCTION

The performance constraint is a probabilistic delay constraint on the end-to-end delay for routing a packet. The end-to-end delay constraint is specified in a similar manner as the hop-to-hop constraint was specified above (in the section on sleeping discipline). In other words, the end-to-end delay is a tuple $\{t, k\}$ and is met when, with probability k, the delay is less than t, *i.e.*, $P[delay > t] < k$.

In the simplified topology used in this paper, the end-to-end delay is a sum of the h hop-to-hop delays. The hop-to-hop delay is comprised of two components. First, there is a delay before a neighboring node wakes up and is available to forward a packet (this is a random quantity, dependent on the sleep discpline). The second hop-to-hop delay component is the time taken to establish the connection between two nodes and transfer the packet (this is a deterministic quantity) Let:

- a_I be the delay before a neighbor is available at the i^{th} hop
- F be duration of a packet transmission.

Thus, writing the end-to-end delay as a sum of the hop-to-hop delays:

$$delay = hF + \sum_{i=1}^{h} a_i$$

This leads to the final constraint equation

$$P\left[hF + \sum_{i=1}^{h} a_i \le t \right] \ge k \qquad (2)$$

CONSTRAINED OPTIMIZATION

Combining the cost function and the constraint function, we obtain a constrained optimization problem. The optimization variables are the number of hops h and the hopping distances, d_i[1]. The optimization problem can be written as follows:

$$\begin{cases} \min_{\{h,d_1,d_2,\ldots,d_h\}} E_{tot} = \min_{\{h,d_1,d_2,\ldots,d_h\}} \\ \left[T\left(l\left(hR + \sum_{i=1}^{h} r d_i^{\,b} \right) + \frac{1}{p}\sum_{i=1}^{h} m_i \left(E_{Id} + g d_i^{\,b} \right) \right) \right] \\ s.t. \\ P\left[hF + \sum_{i=1}^{h} a_i \le t \right] \ge k \\ \sum_{i=1}^{h} d_i = D \end{cases} \qquad (3)$$

In [6] it is shown that (3) is often a convex problem and, even when it is not convex, can usually be solved in a few steps. It is also shown in [6] that the optimum parameters for the sleeping discipline (m_i) are determined by solving (3). Simulation results validating the cross-layer optimization are presented in the next section.

[1] The assumption is that the radios can vary their transmission power to accommodate a range of distances d_i. Alternatively the optimization can be computed before the network is installed and will then impact the optimum network topology for radios with fixed transmission radii.

MODEL VALIDATION

To evaluate the cross-layer optimization model we first solved the optimization problem (3) for a particular network using the Matlab Optimization Toolbox to obtain a theoretical optimum. Then we implemented the same network in OMNeT++ [7] and simulated the performance of this network across a range of parameters to find an empirical optimum. The theoretical and empirical optimums were then compared to evaluate the validity of the model. Note that while the theory was derived using a one dimensional linear network model, the simulations actually used a two dimensional network with nodes having a circular radio range.

The results are presented in Table 1 and Table 2 for a range of network sizes and delay constraints. As can be seen, the optimum values of h obtained through simulation matched the theoretical values (obviously, the number of hops for simulations is restricted to integers).

Table 1. Optimum number of hops and hop-distances for different network sizes and delay constraint of 5seconds

Radius (meters)	h (Theory)	h (Simulations)*	d_i (Theory) -meters	d_i (Simulations) -meters
30	6.41	6	11.03	11.78
40	8.04	8	11.73	11.78
50	9.64	10	12.23	11.78
80	14.38	14	13.10	13.43
100	17.54	18	13.43	13.10

Table 2. Optimum number of hops and hop-distances for different delay constraints and network size of 50m.

Delay Constraint (seconds)	h (Theory)	h (Simulations)*	d_i (Theory) - meters	d_i (Simulations) -meters
3	9.19	9	12.82	13.10
5	9.64	10	12.23	11.78
8	10.16	10	11.59	11.78
10	10.45	10	11.26	11.78

We also simulated the effect of changing the wakeup rate on the power consumption and per-hop delay in the network. 100 nodes were randomly placed in a 40m x 40m area. Each node had a range of 10m. Traffic was generated at all nodes and every packet randomly picked a destination node. The traffic generation rate was Poisson with a mean of 50 seconds. As can be seen in Figure 4, there is a tradeoff between the energy consumption and the delay, which gives the optimal wakeup rate. Note that the plot shows the wakeup rate per node, rather than per region, which was what was derived in the protocol optimization section. However, each forwarding region had 10 nodes on average, so one can easily go from one to the other.

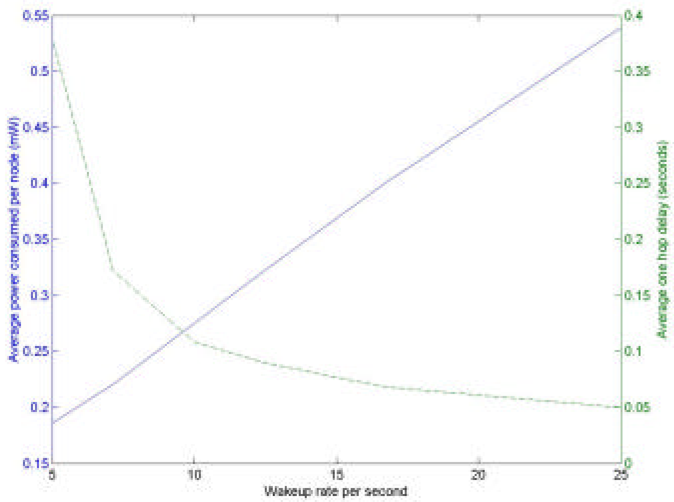

Figure 4 Average power consumption and delay for different wakeup rates

CONCLUSION

This work presents a unified model for the protocol stack. Three algorithms, routing, MAC, and sleep discipline, are developed. These algorithms are at different layers of the stack and are then optimized jointly. We demonstrate that the theoretical optimization solution converges to the empirical optimal solution. We believe that joint optimization is a crucial tool in designing the sensor network protocol stack when energy efficiency is imperative. Joint optimization enables additional efficiencies to be gained by optimizing the interaction between layers. We also introduce the concept of utilizing the randomness present in the network environment to improve network operation. The routing discipline utilized randomness in picking the next-hop neighbor so that energy is spent on a transmission only when there is a high chance of success. The MAC scheme, TICER, enables this type of opportunistic routing to occur. Finally, the sleeping discipline utilizes randomness in the sleeping-times of nodes to ensure that the activity in a region is statistically high enough to support the required performance without adding a communication overhead to ensure coordination between nodes. In conclusion, this paper demonstrated the utility of incorporating statistical randomness and cross-layer design into the sensor network protocol stack.

REFERENCES

[1] D. Tse and S. Hanly, “Multi-access fading channels: Part I: Polymatroid structure, optimal resource allocation and throughput capacities”, IEEE Trans. on Information Theory, vol. 44, no. 7, Nov. 1998.
[2] R. Jain, A. Puri and R. Sengupta, “Geographical routing using partial information for wireless ad hoc networks”, IEEE Personal Comm. Mag., Feb. 2001.
[3] R. C. Shah, S. Wietholter, A. Wolisz and J. Rabaey, “Region-based opportunistic routing for sensor networks”, submitted to ACM Sensys 2004.
[4] E. Lin, A. Wolisz and J. Rabaey, “Power efficient rendezvous schemes for dense wireless sensor networks”, IEEE ICC, June 2004.
[5] J van Greunen, D. Petrovic, A. Bonivento, J. Rabaey, K. Ramchandran, A, Sangiovanni-Vincentelli. “Adaptive Sleep Discipline for Energy Conservation and Robustness in Dense Sensor Networks”, IEEE ICC, June 2004.
[6] A. Bonivento, “Energy efficient protocol design for sensor networks”, M.S. Thesis, UC Berkeley, 2004.
[7] A. Varga, “The OMNeT++ discrete event simulation system”, European Simulation Multiconference, June 2001.