

CROSS-LAYER PROTOCOLS FOR ENERGY-EFFICIENT WIRELESS SENSOR NETWORKING

Brian DeCleene, Victor Firoiu, Matthew Dorsch, Steve Zabele
{brian.decleene, victor.firoiu, matthew.dorsch, steve.zabele}@baesystems.com
Advanced Information Technologies
BAE Systems
Burlington, MA 01803

ABSTRACT¹

The Army's next generation distributed sensor networks provide stand-off situational awareness for future troops. A major factor for the performance and lifecycle of these sensor networks is their ability to conserve battery power. As advances in RF components reach the physical limits of energy efficiency, new network protocols operating at the link layer and above hold the greatest opportunities for additional improvements in energy efficiency. Recently, a number of power-saving solutions have been proposed that separately consider power-consumption of Media Access Control (MAC) scheduling and routing algorithms, but have not considered the potential benefit of cross-layer optimization across these algorithms.

In this paper, we present an approach that integrates pseudo-dynamic scheduling at the link-layer with diversity routing at the network-layer. The link-layer protocol also provides detailed control of the physical-layer's radio state. Our work focuses on reducing the energy loss due to idle listening, control signaling, congestion hot-spots, and packet collisions. We present analytical and simulation results that demonstrate the increased energy efficiency of this class of cross-layer protocols, examine the throughput and latency impacts and define how parameters should be set to optimize the end-to-end performance. We anticipate that the technology presented here will have broad application to army sensor networks as well as public commercial wireless systems and plant automation.

1. INTRODUCTION

The Army's next generation distributed sensor networks provide stand-off situational awareness for future troops. In operations, individual sensors exchange information using battery-powered, wireless data communication links to perform services such as target tracking and then disseminate this information to the appropriate manned control units for decision making. While the node density of the sensor field is sufficiently high to ensure that each node can communicate with one or more neighbors, these fields may extend over very long ranges (on the order of 35km) and therefore require ad-hoc routing algorithms to exfiltrate the data, and media access control (MAC) proto-

cols to coordinate access to the shared wireless channels. The operational effectiveness of a sensor network ultimately depends on the reliability, timeliness, and lifecycle of the deployed system.

A major factor for the performance and lifecycle of a sensor network is its ability to preserve battery power. For example, published requirements for the Advanced Remote Ground Unattended Sensor (ARGUS) Unmanned Ground Sensor (UGS) system requires 90 days of maintenance free operations. With advances in material science and device fabrication, power consumption by the on-board processor and sensor elements have been significantly reduced. As a result, recent analyses on programs such the Massively Deployed Unattended Ground Sensor (MDUGS) and Small Unit Operation/Situation Awareness System (SUO/SAS) have shown that communications and networking account for the majority of the system's power consumption. Figure 1 illustrates some of the opportunities for eliminating the major sources of energy inefficiency at different layers within the OSI protocol stack. Consequently, new network protocols operating at the link layer and above hold the greatest opportunities for additional improvements in energy efficiency.

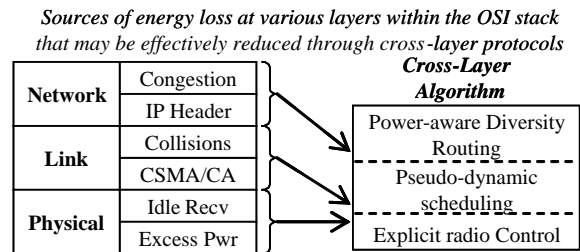


Figure 1: Energy gains by removing cross-layer inefficiencies

Recent studies such as [1] have identified the most important sources of energy loss. **Idle listening**, such as listening to detect start of a neighbor's transmission and processing the header to determine if it is addressed to the node, has been shown in [3] to account for 33-50% of the energy consumed while not transmitting. The transmission and reception of **control packets** such as RTS/CTS/ACK was shown in [2] to account for up to 40% overhead in

¹ This material is based upon work supported by the United States Army Research Office under Contract No. W911NF-04-C-0140 and it does not necessarily reflect the position or policy of the US Government and no official endorsement should be inferred.

CSMA/CA sensor networks with short packet size. Loss of packets due to **collision** requires retransmission, wasting additional energy.

As a result, researchers have begun to examine techniques to synchronize the off-cycles of the radios and thereby reduce the time spent in idle listening [2][3]. Yet, during the on-cycles, the problems of control signaling, errant listening, and collisions reoccur and continue to limit subsequent energy reductions since the data transmissions period (when all but the intended sender/receiver nodes return to sleep) is constrained by the physics of the topology and not the protocol. Thus, to reduce packet collisions, channel partitioning and out-of-band signaling have been recently proposed [4][5]. Within the DoD, this approach is being examined as part of a next-generation sensor package with integrated communications.

Another important source of energy loss is at the network layer. Under current algorithms based on RIP or OSPF, each node routes traffic through the same “optimal” path. As a result, certain nodes become **network hot-spots** and handle an inordinate amount of traffic because they are along a common shortest path for many nodes. When these congested nodes fail as a result of the increased demand, end-to-end performance drops dramatically and the network may partition into two isolated subnets.

These sources of energy inefficiency are the result of the OSI layered paradigm that is ideally suited for the heterogeneous, federated environment of the Internet. For DoD sensor networks, a new cross-layer approach can provide substantial gains in energy efficiency by sacrificing layer independence. In this paper, we replace layering with the assumption that 1) sensor elements share a common protocol suite; 2) computational complexity (processing and memory) is inexpensive compared to transmission costs; and 3) detailed control across layers is allowed. Our research further leverages the ability to control the state of the radio (i.e., transmit, receive/listen, or sleep) for additional energy efficiency.

In this work, we first consider the problem of increasing energy efficiency at the physical and link layers. We define a scheduling algorithm (EPARS) that determines a pseudo-random schedule of the neighbors and hidden nodes. Given that any node can compute its neighbors’ schedules, idle listening is minimized and signaling to coordinate channel access is eliminated. We provide a method for optimal selection of the send probabilities given a minimum desired idle time and the average geographical density of the sensor nodes. We evaluate the performance of our scheduling algorithm against two limiting cases of static channel partitioning algorithms. We find that EPARS offers significant opportunities for trading-off

power for throughput and does not require a-priori network knowledge to configure the system.

Next, we consider the problem of optimizing routing for power efficiency. Following our Cross-Layer approach, we combine link availability information from EPARS with topology and per-link energy consumption to derive a routing algorithm that exploits multiple routes to increase throughput and balance energy usage among the sensor nodes. The routes are the solution of a constraint-based optimization problem solved through linear and quadratic programming. Through simulation, we find that the routes computed by this algorithm, when integrated with EPARS, significantly improve both system lifetime and latency.

Approaches for cross-layer optimization have received significant attention within the sensor networking community. Techniques for joint optimization across the physical and link-layers in concert with other OSI layers have been proposed ranging from utilization of route information [15], [16] to transport congestion control [17]. Correlated research within at the link-layer community has focused on approaches to negotiate transmission schedules to maximize sleep cycles [18] which may need to be renegotiated when the network topology changes due to mobility. Other approaches investigate the construction of transmission schedules that reduce radio interference and distributed power control that satisfies single-hop transmission requirements [19]. Our contribution reduces control signaling at the link-layer by optimizing the probabilistic schedules from information gathered at the routing layer (i.e., two-hop neighbor connectivity) and then feeding this information to the routing layer to optimally weight the traffic flows based on estimates of the link availability. By design, each node can deterministically derive the schedule of neighboring nodes without additional information.

The remainder of the paper is organized as follows. In Section 2 we present the EPARS scheduling algorithm and evaluate its performance. In Section 3 we consider the power-optimal diversity routing algorithm and its performance when combined with EPARS. In Section 4 we discuss our implementation on the MicroMote platform and conclude in Section 5.

2. POWER AWARE SCHEDULING

Random-access MAC protocols (such as CSMA/CA) provide on-demand access to the shared wireless resources and operate independently of the sensor distribution. Each node generally competes for the channel using a combination of request-to-send (RTS) and clear-to-send (CTS) messages. Unfortunately, this comes at an increased energy cost due to signaling (typically 40% of total transmission) and idle listening (33-50%).

On the other hand, channel partitioning schemes, such as TDMA, offer a solution to avoid collisions without additional energy loss due to control signaling. Here, the channel is divided into non-overlapping transmission opportunities which are assigned to the various nodes. A tradeoff is that, as the number of nodes competing for the channel increases, the allocation of partitions becomes increasingly difficult. Solutions such as node grouping in clusters and hierarchical communication through cluster heads add significant overhead and prohibitive energy usage.

Power Aware Random Scheduling

To achieve energy efficiency comparable to channel partitioning while having the flexibility of random access MAC protocols, we propose a pseudo-dynamic MAC scheduling starting from a recent work by Shepard [6]. In the Power-Aware Random Scheduling (PARS) MAC algorithm, each node generates a sequence of time intervals. In each such interval, the node can be in one of four states: send, receive, idle (radio off), or sleep (radio and CPU off.) The sequence of interval durations and associated states, called a schedule, is computed locally by a pseudo-random number generator using a seed based on the node's unique ID number. A node, after learning its neighbors' IDs, can compute locally their schedule by using the same pseudo-random algorithm. Thus, a node can determine the next transmit opportunity to any of its neighbors as the time when its own schedule is in send mode and the neighbor's schedule is in receive mode.

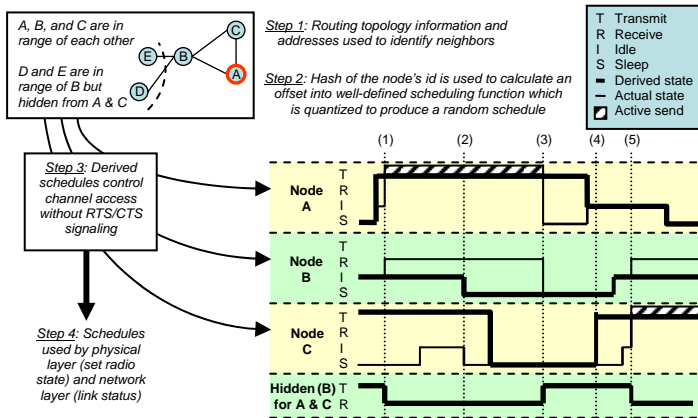


Figure 2: A pseudo-dynamic schedule

Observe that this algorithm is based on a configured set of per-state probabilities, and thus, the probability for any two neighbors being able to communicate is configurable as well. Another issue is that, when using a common pseudo-random number generator (PRNG) for schedule generation, there is a possibility that neighboring nodes generate random stream fragments that prevent successful bi-directional communication for the fragment duration. This possibility can be minimized by using PRNGs that

exhibit good spectral characteristics [12], [13] and a modulus that exceeds the number of output values by several orders of magnitude [14]. The choice of the modulus must balance its performance with the computational capability of the platform (sensor node).

Other issues such as transmit collisions are also resolved based on the pseudo-random schedules. For example, consider a network of five nodes, A to E as in Figure 2-upper. Node A has two direct neighbors, B and C along with two hidden nodes, D and E, which may interfere with channel access. In Figure 2-lower, we show a sample set of schedules for nodes A through E. At time (1), node A can transmit to node B since it is in idle mode but cannot send to node C which is in transmit mode. To resolve conflicts between multiple nodes having overlapping send schedules (such as nodes A and C), we use passive sensing of channel combined with precedence rules. In addition, each node can generate a schedule for the hidden terminals. In Figure 2-lower, this is shown as the bottom line for the terminals hidden from nodes A and C behind B. Since only transmitting nodes represent a potential collision, the hidden node's schedule can be aggregated. In this case, a transmission from node A to node B is delayed to time (1) when none of the hidden nodes are potentially in transmit mode.

Extended Power Aware Random Scheduling

Unlike strict channel partitioning approaches, in our extended scheduling algorithm (EPARS) nodes may deviate from the schedule under certain circumstances to save power or increase throughput. Returning to Figure 2-lower, at time (1), node A initiates a collision-free transmission to node B causing node B to transition to its receive state, thus saving energy by eliminating MAC signaling. Since node C has no information to send, it goes into sleep mode (saving energy through proactive sleep) even though the derived schedule is transmit. A short time later, node C wakes up with a packet to send to node B. Going first into idle mode, node C senses that the channel is in use and does not start transmitting. At time (2), since A and B are in an active transmission and there are no possible collisions either hidden nodes or peers within its vicinity, node B remains in receive mode rather than entering its sleep state, thus increasing throughput and reducing latency by extending the receive/transmit state under certain conditions. At the same time, node C drops prematurely to sleep mode again since node B's derived schedule suggests that it is in sleep mode. Later, the collision avoidance algorithm causes the transmission to stop at time (3) since a hidden node may be generating a competing transmission at this time. Then, at time (4), node C may enter transmission mode but stays in sleep since it notes that node B is still in sleep mode and a hidden node may con-

flict. Finally, at time (5), node C begins transmitting to node B.

We observe that our approach of combining Layer 3 topology information (which is not used in traditional approaches) with the MAC algorithm provides several significant benefits. EPARS saves energy by avoiding collisions (and thus retransmissions) and without energy-consuming control signaling. Turning radio off while no neighbor is in send mode saves additional energy. The algorithm can be tuned to a desired tradeoff between latency and power through its duty cycle parameters. Last, the algorithm can be easily executed on reasonable processors and operate in real-time.

Performance Evaluation

We are interested in evaluating several key features of our cross layer protocols and compare them with limiting baseline TDMA-like algorithms in terms of energy use and throughput. Specifically,

- For a given scheduling algorithm, a node consumes the minimum amount of power when it has no data to transmit or receive. It consumes power only during the scheduled receive intervals. By normalizing it with the power necessary in receive mode, we define the *minimum quiescent energy* E_{min} as the fraction of time spent in receive mode.
- Another key metric is the maximum throughput of a node over a long period of time. By normalizing this throughput with the bandwidth of the wireless link, we define the throughput ρ to be the fraction of time that the node is able to use the available channel to send data.

For these metrics, we define two fair transmission scheduling algorithms, Maximum Throughput (A1) and Minimum Receive Energy (A2) for comparative analysis of the EPARS algorithm in the case of a single hop network configuration.

A1) In the Maximum Throughput algorithm, each node is assigned a transmission slot of duration $1/N$. When not sending, each receiver must listen to the channel to detect any transmissions from one of their neighbors. It follows that

$$\rho^{A1} = 1/N \quad E_{min}^{A1} = \frac{N-1}{N}$$

A2) The Maximum Lifetime algorithm minimizes the energy consumed by each node by assigning to each source/destination pair a separate time slot. During slots that are not assigned to a particular node, the node may sleep to reduce the power consumed. It is easy to see that

$$\rho^{A2} = \frac{1}{N(N-1)} \quad E_{min}^{A2} = 1/N$$

In the EPARS scheduling algorithm, a node is in the send, receive, or idle state with probability φ_s , φ_r , and φ_i respectively. A send opportunity occurs when the sender is in send-mode and the receiver is in receive-mode. Since the EPARS algorithm allows overlapping send states to be used and send conflicts to be resolved by channel sensing, we define κ to be the probability that a transmission is blocked during such send state overlap. From these definitions, the throughput and minimum energy drain in EPARS are given by

$$\rho^{RD} = \varphi_s \varphi_r [1 - \kappa \varphi_s]^{N-2} \quad E_{min}^{RD} = \varphi_r$$

In EPARS, φ_s and φ_r are configured by optimizing the throughput for a fixed amount of idle time φ_i . By increasing the amount of idle time, we can reduce the power consumed. Similarly, by decreasing the idle time, we can increase a node's potential throughput.

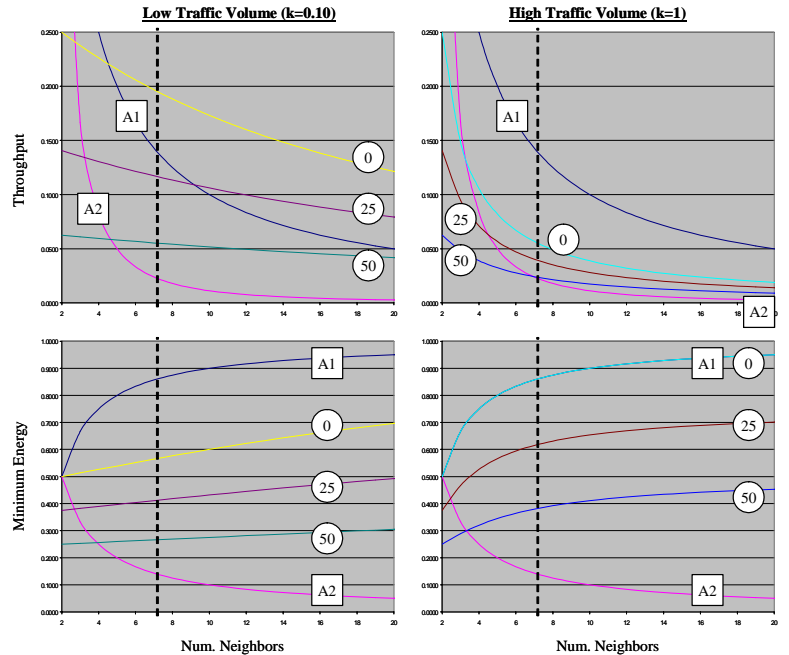


Figure 3: Comparison of Throughput and Minimum Energy Consumption for EPARS and Static Channel Partitioning Algorithms

Figure 3 shows the performance of the static channel partitioning algorithms, A1 and A2, along with a basic random scheduling algorithm with different fractions of idle time (i.e., 0%, 25%, and 50%) and with different traffic loads ($\kappa = 0.10$ and $\kappa = 1$). The approximate network percolation point is shown at the dotted vertical line and corresponds to the minimum node density required to achieve a fully connected network over a distributed field of sensors.

When the random scheduling algorithm is able to tolerate a relatively high amount of potential conflict or has low traf-

fic load (i.e., $\kappa = 0.10$), the random scheduling algorithm achieves higher throughput than the maximum throughput algorithm (A1) at a lower energy level. This is a consequence of the algorithm’s ability to resolve overlapping send opportunities across different nodes. As the level of tolerable competition decreases (i.e., $\kappa \rightarrow 1$), the throughput of the random scheduling algorithm decreases while the energy consumed increases. Although the marginal benefit over the (A2) algorithm decreases as well, the throughput of the random scheduling algorithm is consistently higher than the (A2) algorithm for reasonably-sized node densities. By changing the idle time, we can select an energy/throughput point that matches the desired performance characteristics. Thus, random scheduling allows the system to trade-off performance versus energy while simultaneously not requiring a-priori schedule coordination.

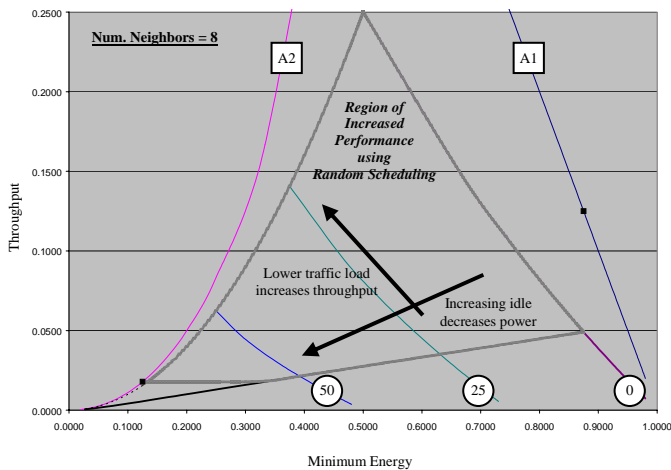


Figure 4: Trade-Space Diagrams of EPARS and Static Channel Partitioning Algorithms show regions of increased performance for the random-scheduling algorithm

The energy/throughput trade-offs are shown in Figure 4. The operating points for the Maximum Throughput (A1) and Maximum Lifetime (A2) are marked along their respective energy/trade-off curves. As we can see, there is a region within the random-schedule trade-off that provides improved performance for a given minimum energy load. By increasing the idle period, the random schedule can reduce the minimum energy. Similarly, decreasing κ improves throughput. While these parameters are non-orthogonal, they allow the system to select an operating point tailored for its intended use.

So far we have modeled the performance of EPARS scheduling in a simple network where all nodes are within transmission range. We have also modeled EPARS performance in a general network where we take into account possible scheduling conflicts between transmit opportunities of nodes that are two hops distant (i.e., have a common neighbor but are not in direct range). Using a two-

dimensional Poisson distribution of nodes, we derived tight bounds on the probability of hidden node interference and computed the optimal send probability for configuring the EPARS schedule.

3. EPARS WITH DIVERSITY ROUTING

In developing EPARS, two-hop neighbor information from the network layer was required to perform the power and throughput optimization. A similar exchange of link-layer schedule information to the network-layer can further improve performance. The central concept is to utilize multiple potential send epics to different next-hop neighbors in order to 1) maximize throughput of the system; 2) decrease latency; and 3) eliminate traffic hot-spots which suffer an inordinate amount of energy consumption.

Traffic hot-spots occur in most ad-hoc network and classical dynamic routing protocols due to their tendency to select and use a small number of paths, which become heavily used [8]. As a result, overuse of common paths unfairly drains the energy of nodes along this “optimal” path. Consequently, nodes along this path may prematurely fail due to forwarding demands – creating gaps in the sensor field and fragmentation of the network topology. Furthermore, as multiple nodes select common routes, congestion along those routes increases. Given that nodes spend a majority of their time in a communications sleep (i.e., quiescent) state to save power, congestion latencies can become significant as the congested node is forced to wait for appropriate transmission rounds while transmission opportunities from other nodes sit idle.

The class of diversity routing protocols provides a solution to energy-draining hot-spots and node congestion. Under diversity routing, nodes set up, maintain, and simultaneously use multiple, parallel paths across the network per individual link characteristics. This offers load balancing and the potential for optimizing network utilization and efficiency as well as a very viable means for achieving nearly instantaneous local failover when individual network nodes or links fail.

The core for our cross-layer routing approach derives from a multi-commodity flow formulation that is solved through a sequence of constrained linear (LP) and quadratic (QP) programming techniques. Under this formulation, the EPARS successful send probability is incorporated into the optimization function. Constraints on energy flow and system lifetime work in concert to distribute the load and avoid hot-spots. We briefly outline the formulation below.

Assume that each node i has finite initial battery energy of E_i and generates traffic of commodity c with an intensity of $Q_i^{(c)}$. For each neighbor, j , we define $e_{i,j}$ as the energy consumed to transmit an information unit to j ; and

the flow of commodity c between these two nodes as $q_{i,j}^{(c)}$. These flows represent the weights to forward traffic to each of the next-hop neighbors. To capture the effect of hot-spots in the network, we introduce an additional variable, T_{sys} , which is the duration until the first node in the system fails due to energy drain. Hot spots in the network result in accelerated energy loss and a lower system time. Finally, we define the amount of a particular commodity that is transmitted until the first node failure as $\hat{q}_{i,j}^{(c)} = T_{\text{sys}} q_{i,j}^{(c)}$. Under this formulation, we define the following constraints:

- The lifetime of the system and all flows must be positive (i.e., $T_{\text{sys}} \geq 0$ and $\hat{q}_{i,j}^{(c)} \geq 0$).
- The total energy consumed by a node must be less than the total available energy of the node. That is,
$$\sum_j e_{i,j} \sum_c \hat{q}_{i,j}^{(c)} \leq E_i$$
- For each commodity, the total outbound traffic from each node equals the total incoming traffic that is forwarded plus the traffic generated by the node.

Observe that no bandwidth constraints have been introduced. For example, bandwidth constraints may be defined along individual links within the topology. Alternatively, within a wireless environment we may define constraints that bound the total amount of traffic that the node can handle due to the fact that a single transceiver cannot send and receive at the same time.

The cross-layer routing tables are constructed as follows:

- 1) Calculate the next-hop weights peak achievable system lifetime by maximizing T_{sys} subject to the constraints above. Denote the solution to this as $T_{\text{sys}}^{(1)}$.
- 2) Calculate the shortest number of total hops for the traffic by minimize $\sum_{i,j,c} \hat{q}_{i,j}^{(c)}$ subject to $T_{\text{sys}} \geq \alpha T_{\text{sys}}^{(1)}$ and the earlier constraints where α is a control parameter between 0 and 1. We denote this solution as $L_{\text{sys}}^{(2)}$.
- 3) Minimize the quadratic system $\sum_{c \in C} \left(\hat{q}_{i,j}^{(c)} / a_{i,j} \right)^2$ subject to $T_{\text{sys}} \geq \alpha T_{\text{sys}}^{(1)}$, $\beta \sum_{i,j,c} \hat{q}_{i,j}^{(c)} \leq L_{\text{sys}}^{(2)}$, and the earlier constraints. The parameter $a_{i,j}$ is the probability of a collision free send opportunity between nodes i and j derived from the optimal EPARS formulation discussed in the previous section.

We emphasize the use of link availability a_{ij} in the route optimization problem. This Cross-Layer optimization has benefits above single-layer approaches, as shown below.

By solving the first LP problem, our algorithm identifies the optimal forwarding solution to avoid hot-spots in the network. Using this as a constraint, we then construct the Dijkstra-like minimum hop solution that is “near” this optimal lifetime solution. Controlled by the parameter α , this allows us to trade between the competing power demands of routing hot-spots (which excessively drains certain nodes) and long-paths (which consume more energy by virtue of the increased number of hops).

While the first two LP formulations allow us to identify feasible solutions that trade-off competing energy demands, they do not generally result in multipath solutions. This is due to the fact that efficient LP solvers search for solutions by considering the vertices of the solution space and conclude once a solution is found. By virtue of examining the vertices of the solution space, all traffic for a particular destination tends to be directed along a single next-hop. The final QP optimization introduces an incentive to examine other equally valid solutions that have maximal diversity as well. That is, solutions that distribute the load across multiple equivalent paths reduce the total QP cost.

4. IMPLEMENTATION and EVALUATION

To examine the characteristics of the EPARS/PALM algorithm, we examined both the general analytical characteristics of the algorithm during the first phase and the implementation specific details during the second phase. In a first evaluation phase, we have developed a simulation environment based on the analytical models presented in the previous sections for the EPARS/PALMS, Maximum Throughput and Maximum Lifetime algorithms. In the following, we present results from a simulation of a network with 100 nodes distributed across the plane using a 2D random Poisson distribution. The transmission range of each node was set to yield an average of 10 neighbors per node. In the top row of Figure 5 we show the route selected for each algorithm between a fixed pair of nodes. As expected, the Maximum Throughput solution (derived from the second step above with $\alpha=1$) selects the shortest path to the target destination. The Maximum Lifetime (derived from the first step above) solution selects a more circuitous route in order to balance the traffic loads and extend the network’s lifetime. The combined algorithm produces a route that is slightly longer than the shortest path. It also favors nodes where the node density is lower, yielding greater opportunities to send data collision free.

The different degrees of path diversity constructed by each algorithm are shown in the second row of the figure. Thin links have no path diversity while thick links carry at least

a small fraction of traffic that is divided with another link. The Maximum Throughput and Maximum Lifetime solutions consist almost entirely of unipath routes (A close examination of the Maximum Lifetime solution shows a couple multipath routes in the lower left corner). The integrated PALMS/EPARS algorithm, on the other hand, exhibits significant path diversity.

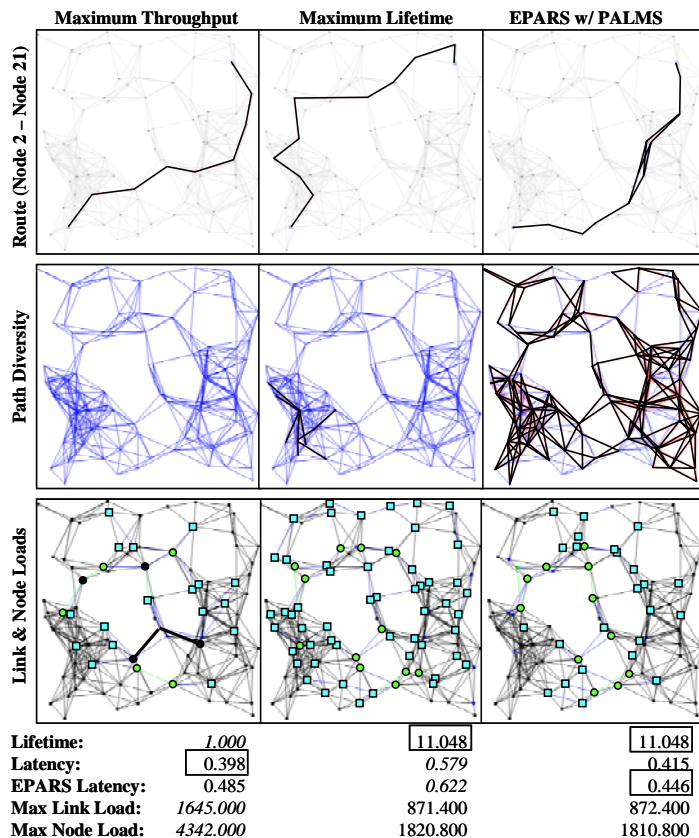


Figure 5: Comparison of Routing Formulations

The third row of Figure 5 shows the traffic and node loading for each of the three algorithms. By optimizing the latency, the Maximum Throughput solution results in a couple of heavily loaded links (thick lines) and heavily utilized nodes (denoted by filled circles). These load values are also listed in the bottom row and show that the Maximum Throughput formulation has more than double the load on key links and nodes. In the maximum latency solution, nearly every node is used to balance the traffic. This can be noted by the even distribution of modestly loaded square nodes throughout the entire topology. The integrated algorithm results in similar node and link loading with the bulk of the traffic distributed in the middle of the topology and very little traffic near the network edges.

Finally, the bottom row contains the performance figures for the various algorithms with the best value for each metric marked by a box and the worst value italicized. One observation is that the Maximum Lifetime and EPARS

algorithms all achieve the maximum system lifetime. The general latency (assuming instant access to the channel with no collisions) is higher for the EPARS algorithm but below the Maximum Lifetime solution. However, when the link-layer scheduling information is included in the routing computation, the resulting cross-layer power-aware algorithm performs optimally.

In a second phase, we have implemented our algorithms onto the MicroMote platform using the TinyOS operating system [10], an open source operating system for micro-sensors developed at Berkeley. We developed our code for deployment in sensor networks and it is currently under test in our lab using Crossbow Mica2 and Mica2dot hardware. Given limitations in available sensor hardware, we evaluated the system’s scalability using the TinyOS Simulator as well as demonstrating its operation on the Mote platform. The simulator was instrumented with an estimator of power consumed that uses the measured time spent in different states (send, receive, sleep) and the average power consumed in each of these states (measured in mA, assuming constant voltage).

In the following, we present results from a set of experiments involving 100 TinyOS motes randomly placed in a plane with each mote having an average of 10 neighbors. The application on each mote generates 90 packets per hour of sensor traffic destined for a randomly chosen destination. Over the course of the experiment, each mote reported state, power, and communications data. The capacity of the battery was set to 0.8 mAh and the simulation was run for two hours.

Due to the low traffic data rates, both Maximum Throughput and Maximum Lifetime algorithms presented previously performed very poorly in terms for energy efficiency due to idle listening. We therefore introduced a system-wide sleep cycle to these algorithms such that all of the nodes simultaneously slept for a fraction of the total operating time.² In the following experiments, we set the sleep time for EPARS, Maximum Throughput and Maximum Lifetime to 92%, 80% and 44% respectively, to achieve an average throughput of 1.6 packets/second for all of the algorithms.

Shown in Table 1, we observe that EPARS has a better average current consumption than Maximum Throughput and close to Maximum Lifetime, and outperforms both TDMA-like algorithms in peak current consumption. This results in EPARS having a significantly improved Time-to-First-Node-Failure, a main goal of our optimization.

² Control signaling required to maintain clock synchronization and coordinate schedules was included in EPARS/PALMS but neglected in the modified Maximum Throughput or Maximum Lifetime algorithms.

Also observe that EPARS, compared with the other TDMA schemes, does not need Global Time synchronization nor signaling for scheduling states, which represents a significant amount of additional power saved.

Table 1: Performance comparison, TinyOS Implementation

	Maximum Throughput	EPARS w/ PALMS	Maximum Lifetime
Idle (%)	91.52	97.26	96.41
Send (%)	0.86	1.34	2.12
Receive (%)	7.62	1.37	1.47
Ave Current (mA)	1.380	0.403	0.395
Maximum Current (mA)	2.663	0.606	0.688
Ave Pkts Delivered (p/s)	1.78	1.57	1.58
First Node Failure (h:mm)	0:17	1:12	1:04
Needs Global Time	Yes	No	Yes
Needs Signaling	Yes	No	Yes

5. DISCUSSION

The EPARS algorithm requires accurate time synchronization between any pair of neighboring nodes. We implemented the Reference Broadcast System algorithm [11], as well as a Sender-Receiver method. We chose the latter due to its smaller communication overhead and better accuracy. We are currently investigating tradeoffs between time synchronization accuracy and the frequency of clock synchronization messages (timestamps) in the context of minimizing the power consumption. We are also examining computationally efficient approximations of the PALMS routing algorithm for improved performance.

6. CONCLUSION

Cross-layer protocols hold significant promise for reducing the power consumption within sensor networks. Major sources of energy loss can be countered by taking advantage of information available at other layers in the network. This work illustrates how the combination of link-layer control of the radio's power mode combined with network-layer diversity routing can significantly increase the lifetime of a single node as well as the entire network. Combined with other advances in the areas of low-power sensing and computing, future sensors should be able to substantially increase their operational lifetime before requiring maintenance. As the military continues to move toward network-centric operations that limit the exposure of military personnel to harm, advanced sensor network technologies promise to provide a sustainable military presence in denied or controlled areas.

7. ACKNOWLEDGEMENTS

We would like to thank Jeffrey Opper for many helpful discussions.

REFERENCES

- [1] W. Ye, J. Heidemann, and D. Estrin. "An Energy-Efficient MAC protocol for Wireless Sensor Networks." Technical Report ISI-TR-543, USC/ISI, Sept. 2001.
- [2] A. Woo, D. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks", International Conference on Mobile Computing and Networking, 2001.
- [3] F. Bennett, D. Clarke, et al, "Piconet: Embedded mobile networking," IEEE Personal communication Magazine, vol. 4, no. 5, pp. 8-15, 1997.
- [4] C. Raghavendra, S. Singh, "PAMAS – Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks", ACM Computer Communication Review, 1998.
- [5] G. Girling, J. Li Kam Wa, P. Osborn, R. Stefanova, "The PEN Low Power Protocol Stack", IEEE Int'l Conference on Computer Communications and Networks, 2000
- [6] T. Shepard, "A Channel Access Scheme for Large Dense Packet Radio Networks," ACM SIGCOMM Aug. 1996.
- [7] IETF MANET Working Group Draft, "Ad Hoc On Demand Distance Vector (AODV) routing", Santau Das, C. Perkins, Elizabeth Royer, 2002
- [8] D. Bertsekas, "Dynamic Behavior of Shortest Path Routing Algorithms for Communication Networks," IEEE Transactions on Automatic Control, Feb 1982.
- [9] ALPHATECH Inc., Precision fire control tracking algorithm description document, Technical Report TR-1011, April 2002. www.alphatech.com
- [10] TinyOS Community Forum, <http://webs.cs.berkeley.edu/tos/>, 2004.
- [11] J.Elson, L.Girod, D.Estrin, "Fine-grained network time synchronization using Reference Broadcasts", In proceedings of OSDI 2002.
- [12] R. Coveyou and R. MacPherson, "Fourier Analysis of Uniform Random Number Generators", Journal of the ACM, Vol.14, Issue1, 100 - 119, January 1967.
- [13] K. Entacher, "Bad Sequences of Well-Known Linear Congruential Pseudo-Random Number Generators," *ACM Transactions on Computer Simulation*, January 1998.
- [14] P. L'Ecuyer, "Tables of Linear Congruential Generators of Different Sizes and Good Lattice Structure," *Mathematics of Computation*, Vol. 68, No. 225, pp. 249 - 260, Jan1999.
- [15] M. Sichitiu, "Cross-layer scheduling for power efficiency in wireless sensor networks", *Proc. IEEE INFOCOM*, 2004, pp. 1741 - 1751, March 2004.
- [16] W.Yuen, H. Lee, and T. Andersen, "A Simple and Effective Cross Layer Networking System for Mobile Ad Hoc Networks," *Proc. IEEE PIMRC '02*.
- [17] M. Chiang, "To layer or not to layer Balancing transport and physical layers in wireless multihop networks", *Proc. IEEE INFOCOM*, 2004, pp. 2525 - 2536, March 2004.
- [18] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks", *IEEE/ACM Trans. Netw.*, June 2004.
- [19] T. ElBatt and A. Ephremides, "Joint scheduling and power control for wireless ad-hoc networks", *Proc. IEEE INFOCOM*, 2002, pp. 976 - 984, June 2002.