This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

SolarCode: Utilizing Erasure Codes for Reliable Data Delivery in Solar-powered Wireless Sensor Networks

Yong Yang, Lu Su, Yan Gao and Tarek F. Abdelzaher
Department of Computer Science, University of Illinois at Urbana-Champaign
Email: {yang25, lusu2, yangao3, zaher}@illinois.edu

***Abstract*—Solar-powered sensor nodes have incentive to spend extra energy, especially when the battery is fully charged, because this energy surplus would be wasted otherwise. In this paper, we consider the problem of utilizing such energy surplus to adaptively adjust the redundancy level of erasure codes used in communication, so that the delivery reliability is improved while the network lifetime is still conserved. We formulate the problem as maximizing the end-to-end packet delivery probability under energy constraints. This formulated problem is hard to solve because of the combinatorics involved and the special curvature of its objective function. By exploiting its inherent properties, we propose an effective solution called SolarCode, which has a constant approximation ratio. We evaluate SolarCode in the context of our solar-powered sensor network testbed. Experiments show that SolarCode is successful in utilizing energy surplus and leads to higher data delivery reliability.**

I. Introduction

Because of the error-prone nature of wireless communication, a spectrum of solutions has been proposed at every layer of the network protocol stack on how to reliably deliver the sensory data in wireless sensor networks. Among them, a class of approaches is to proactively add redundancy by using simple duplication or advanced coding schemes (e.g., erasure coding [1], [2]), and send multiple copies of a message simultaneously to mitigate the effects of single-message losses. Due to the limited energy in traditional battery-powered sensor networks, exploiting any level of redundancy will inevitably reduce the network lifetime. This is because achieving redundancy takes extra energy and the total amount of work that can be accomplished by a node is pre-determined by the initial energy in its battery. Therefore, most prior efforts mainly focus on the trade-off between reliability and network lifetime.

However, this dilemma could be relieved in wireless sensor networks with renewable energy sources. Considering the fact that a full battery can not harvest more energy, there is an incentive to spend energy to make room to harvest more energy. As a result, the extra spending has no impact on the node lifetime since this energy surplus would be wasted otherwise. In this paper, we consider the problem of utilizing such energy surplus (if any) in solar-powered sensor networks to adaptively adjust the redundancy level of erasure codes used in communication, so that the data delivery reliability is improved while the network lifetime is still conserved.

As an efficient technique for recovering data from partial loss or corruption, erasure coding has long been adopted for peer-to-peer systems [3] and delay tolerant networks [4], coping with the failure of packet transmissions [5], [2] or the breakdown of storage systems [6], [7]. An erasure coding scheme encodes a message into a large number of blocks. The original message

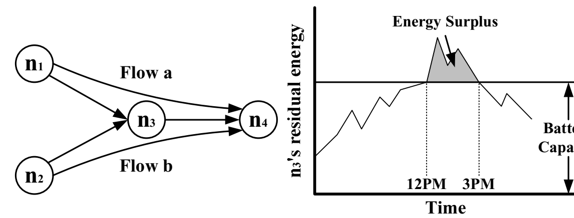

Fig. 1. A simple topology in which two flows a and b, originated from node n_1 and n_2 respectively, go to node n_4 through a relay node n_3. The residual energy of n_3 is shown on the right.

can be recovered as long as enough encoded blocks are received. The more encoded blocks are generated and transmitted, the more likely it is that the original message can be recovered. In the context of sensor networks, the redundancy level should be adjusted dynamically according to the energy availability of sensor nodes.

The uniqueness and challenge of this problem can be easily illustrated by a simple example. Consider the topology shown in Figure 1, which also shows the residual energy of node n_3 in one day. As we can see, n_3 is fully charged around noon and stays in the fully charged state until 3PM because the energy charging rate is higher than the consumption rate during these 3 hours. This makes it possible for node n_3 to spend energy at a higher rate during this period (i.e., increase its communication redundancy) and still remain in the fully charged state at 3PM. As a result, the data delivery reliability is improved and the network lifetime remains intact.

A naive approach for utilizing energy surplus is to add communication redundancy only when the battery is full. However, under this simple approach, erasure coding is only active during the short periods of fully charged state. Consequently, the energy surplus may not be fully utilized because adding more redundancy will have a very marginal gain if the redundancy level is already reasonably high for these periods. Furthermore, from the perspective of end-to-end flows, a node (e.g., n_1 or n_2 in this example) should not arbitrarily increase its transmission redundancy even though it has plenty of energy surplus. This is because it will take the receiving node (e.g., n_3) extra energy to receive the redundant communication. In this example, if link $(3, 4)$ has a low quality, it may be better for n_3 to allocate most of its energy surplus for transmitting on link (n_3, n_4) than receiving on link (n_1, n_3). As a relay node, another problem that it has to face is how to divide its energy surplus for the passing flow a and b such that the performance of the whole network is improved.

This example shows that even for simple topologies determining the optimal redundancy levels can be complex. The

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

optimal levels depend on not only the network properties (e.g., topology and link qualities), but also the solar energy harvesting process. In this paper, we rigorously formulate an optimization problem to determine how to dynamically adjust the redundancy level of each data link over the time period of interest, such that the end-to-end packet delivery probability is maximized and the network lifetime is not affected.

The formulated optimization problem, however, is in general hard to solve because of the combinatorics involved and the special curvature of its objective function. By exploiting special properties of the problem, we propose an effective approximated solution called ***SolarCode***. We prove that SolarCode has a constant approximation ratio. Moreover, we also prove that the combinatoric functions involved in the objective are guaranteed to be concave, as long as the quality of considered links is not too low. Therefore, SolarCode solves the problem by using general convex optimization techniques. We evaluate the performance of SolarCode based on the real setting of our previously published solar-powered sensor network testbed [8]. Results show that SolarCode leads to a higher end-to-end packet delivery probability.

The remainder of the paper is organized as follows. We formulate the problem in Section II, and propose our approximated solution SolarCode in Section III. Then, we evaluate SolarCode in Section IV, and finally conclude the paper by Section V.

II. Problem Formulation

In this section, we formulate an optimization problem that determines the redundancy level of each link in a solar-powered sensor network such that the total end-to-end packet delivery probability is maximized and the network lifetime is conserved.

We consider a network with a node set $\mathcal{N}$ and a link set $\mathcal{L}$. Let $l(i,j) \in \mathcal{L}$ be a directional link from node i to node j where $i, j \in \mathcal{N}$. Sensory data are generated on nodes in a subset $\mathcal{S} \subseteq \mathcal{N}$, and then forwarded through the network to particular destinations. The traffic pattern that we consider in this paper is general so that different data flows could have a single or multiple destinations, depending on the applications. Solely for the sake of presentation, we assume single path communication, and that the route used by a flow is static throughout the time period under consideration. Moreover, we assume that the route of a flow is determined by some other routing module and is not considered as an optimization knob in this paper. Let r_s be the rate of the sensory data generated on node $s \in \mathcal{S}$, and f_s be the route used to forward these data in the network. We write $l(i,j) \in f_s$ to denote that link $l(i,j)$ is on the route f_s.

Communication on wireless links is error-prone. Many wireless link error models have been proposed. We adopt an effective and widely used statistical BER-based (Bit Error Rate) model. The transmission of a packet is successful only when all its bits are received correctly. Thus, the successful transmission probability for a packet of m bits is $(1-p_e)^m$, where p_e is the statistical bit error rate on this link.

When erasure coding is employed, a packet is first divided into b blocks and then encoded into αb code blocks such that if b or more code blocks are received, the original packet can be decoded. The parameter α determines the degree of redundancy and is called the *replication factor*. Denote $p = (1-p_e)^{\frac{m}{b}}$ as the successful transmission probability of each code block. With a replication factor α, αb code blocks are transmitted over a link and we can express the successful decoding probability as a function of α:

$$\widetilde{Pr}(\alpha) = \sum_{k=b}^{\alpha b} \binom{\alpha b}{k} p^k (1-p)^{\alpha b - k}. \tag{1}$$

Note that αb in Eq (1) has to be an integer. We can allow α to be any real number in $[1, +\infty)$, by using a coding module to always generate $\lfloor \alpha b \rfloor$ code blocks, and generate one extra code block with probability $\alpha b - \lfloor \alpha b \rfloor$. Thus, the successful decoding probability function for a general α is

$$Pr(\alpha) = (1 + \lfloor \alpha b \rfloor - \alpha b)\widetilde{Pr}(\frac{\lfloor \alpha b \rfloor}{b}) + (\alpha b - \lfloor \alpha b \rfloor)\widetilde{Pr}(\frac{\lceil \alpha b \rceil}{b}). \tag{2}$$

Each node $i \in \mathcal{N}$ is powered on a rechargeable battery with capacity B_i, and the battery is charged by a solar panel. The solar energy available for harvesting depends on many factors. In section IV, we will elaborate on how we project the available solar energy based on historical solar energy traces and weather forecast information. Now in our formulation, we assume that the solar energy at time t for node i is known as an input, denoted as $S_i(t)$. Let C_i be the CPU power consumed by applications together with system processes running on node i. The wireless radio of node i has a power consumption rate P_i^{TX} for transmitting and P_i^{RX} for receiving. Hence, the power consumption rate for data transmitting and receiving on link $l(i,j)$ when applying erasure coding is $\alpha_{ij}(t)\ P_i^{TX}$ and $\alpha_{ij}(t)\ P_j^{RX}$, respectively, where $\alpha_{ij}(t)$ is the replication factor used for link $l(i,j)$ at time t. Note that $\alpha_{ij}(t)$ is an absolute factor with respect to the raw data packet. Therefore, the overall power consumption rate of node i at time t is:

$$W_i(t) = C_i + \sum_{l(j,i)} R_{ji}(t)\alpha_{ji}(t)\ P_i^{RX} + \sum_{l(i,j)} R_{ij}(t)\alpha_{ij}(t)\ P_i^{TX}, \tag{3}$$

where $R_{ij}(t) = \sum_{s:l(i,j)\in f_s} r_s$ is the total traffic rate of raw data on link $l(i,j)$ at time t.

When a packet is relayed by a node i to a node j, it is encoded into $\alpha_{ij}(t)b$ blocks and all blocks are sent on link (i,j). The packet can be successfully decoded on node j with probability $Pr(\alpha_{ij}(t))$. If successfully decoded, this packet will then be encoded again into $\alpha_{jk}(t)b$ blocks and sent on link (j,k) to its next hop k. Then the end-to-end packet delivery probability for a flow f_s is $D_s(t) = \prod_{l(i,j)\in f_s} Pr(\alpha_{ij}(t))$.

We discretize the time under consideration, T, into N_t slots. Thus, each of duration is $\Delta = T/N_t$. Our objective is to adapt the replication factor $\alpha_{ij}(t)$ of each link $l(i,j)$ such that the end-to-end packet delivery probability weighted by flow rates throughout T is maximized. Let $R_S = T\sum_{s\in\mathcal{S}} r_s$ be the total traffic load. Therefore our objective function is

$$\max \sum_{s\in\mathcal{S}} \sum_{t=1}^{N_t} \frac{r_s\Delta}{R_S} \prod_{l(i,j)\in f_s} Pr(\alpha_{ij}(t)), \tag{4}$$

subject to the following constraints.

First, our goal is to utilize the energy surplus from a renewable energy source to enhance the data delivery reliability. It has to be ensured that the extra energy spending does not

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

affect the network lifetime. Thus, the residual energy of node i, denoted as $e_i(t)$, should satisfy a *non-blackout constraint*:

$$e_i(t) > 0, \quad \forall i \in \mathcal{N}, 1 \leq t \leq N_t. \tag{5}$$

In order to have feasible solutions, we assume that this non-blackout constraint is satisfied when all $\alpha_{ij}(t)$s equal to one. It means that the system does not have blackout originally when erasure coding is not used. This can be enforced in the design of the actual system.

Second, the residual energy of node i at time slot t (i.e., $e_i(t)$) equals to the remaining energy at the last time slot $(t-1)$ plus the solar energy harvested subtracting the consumed energy $W_i(t)\Delta$. Note that the available solar energy $S_i(t)$ may not be fully harvested into the battery because of the battery capacity bound. Namely, $e_i(t)$ should also be bounded by the battery capacity B_i. Thus, we have an *energy evolution constraint*:

$$e_i(t) = \min\{e_i(t-1) + (S_i(t) - W_i(t))\Delta, B_i\}, \forall i \in \mathcal{N}, 1 \leq t \leq N_t. \tag{6}$$

Observe that the objective function is non-decreasing with $e_i(t)$, which means that the more energy is in the battery, the more is the energy surplus, and the higher is the reliability that can be achieved. Therefore, the *energy evolution constraint* is equivalent to the following two linear constraints:

$$\begin{aligned} e_i(t) &\leq e_i(t-1) + (S_i(t) - W_i(t))\Delta, \\ e_i(t) &\leq B_i, \qquad \forall i \in \mathcal{N}, 1 \leq t \leq N_t. \end{aligned} \tag{7}$$

Finally, the initial energy values $e_i(0)$ ($i \in \mathcal{N}$) are given as inputs. Recall that available solar energy values $S_i(t)$ ($i \in \mathcal{N}, 1 \leq t \leq N_t$) are also assumed to be known.

Before showing how to solve the formulated problem, we discuss two issues related to this formulation. First, besides adding redundancy, retransmission is another option for reliable data communication. However, retransmission usually has a longer delay because of a relatively long time-out duration involved. Therefore, we concentrate on using erasure codes for reliable data delivery in this paper. In fact, our approach can be extended to retransmission schemes, where α can be regarded as an upper bound for the number of retransmissions. Second, aside from the extra energy cost, a higher α could also cause higher interference overhead and bring down the link quality. One can formulate a network cross-layer optimization problem to take into account the effect of replication factors on communication interference. However this would lead to a more complex problem with no effective solution. We assume that the redundant traffic load is much lower than the link capacity, and signal interference can be considered to have nearly no impact on link quality. This is usually true for sensor networks, where raw traffic load is low in most applications.

III. SolarCode: An Effective Solution

One difficulty in solving the optimization problem is from the products of probability functions $Pr(\alpha_{ij}(t))$ in Eq.(4). A natural way to bypass this is to use a logarithm function to transform the products of $Pr(\alpha_{ij}(t))$ into summations of $\log(Pr(\alpha_{ij}(t)))$. Maximizing Eq.(4) is equivalent to maximizing its logarithm value. Notice that $\sum_{s\in\mathcal{S}} \sum_{t=1}^{N_t} \frac{r_s\Delta}{R_S} = 1$. Thus we can obtain an lower bound on the objective function by applying the concave property of logarithm functions,

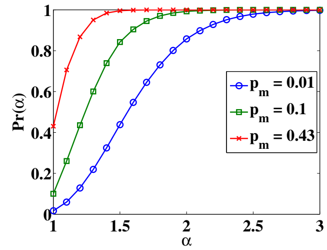

Fig. 2. With $b = 8$, $Pr(\alpha)$ has an inflection point if $p_m = 0.01$, but is concave if $p_m = 0.1$ or 0.43.

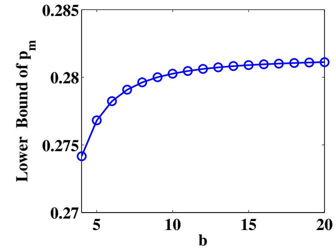

Fig. 3. The lower bound for the packet delivery probability p_m to guarantee the concavity of $Pr(\alpha)$.

$$\begin{aligned} &\log\left(\sum_{s\in\mathcal{S}}\sum_{t=1}^{N_t}\frac{r_s\Delta}{R_S}\prod_{l(i,j)\in f_s}Pr(\alpha_{ij}(t))\right) \\ \geq\ & \sum_{s\in\mathcal{S}}\sum_{t=1}^{N_t}\frac{r_s\Delta}{R_S}\cdot\log\left(\prod_{l(i,j)\in f_s}Pr(\alpha_{ij}(t))\right) \qquad (8) \\ =\ & \sum_{s\in\mathcal{S}}\sum_{t=1}^{N_t}\frac{r_s\Delta}{R_S}\cdot\sum_{l(i,j)\in f_s}\log\Big(Pr(\alpha_{ij}(t))\Big). \qquad (9) \end{aligned}$$

Recall $D_s(t) = \prod_{l(i,j)\in f_s} Pr(\alpha_{ij}(t))$. The equality in (8) holds when all $D_s(t)$s are equal for $\forall s \in \mathcal{S}$ and $1 \leq t \leq N_t$, but there is no general upper bound for the difference between the two sides of this inequality. In particular, it becomes infinity when any $D_s(t)$ becomes infinitesimal. Considering that $D_s(t)$ is the end-to-end delivery probability of a flow, we have $\beta = \inf\{D_s(t) : s \in \mathcal{S}, 1 \leq t \leq N_t\} > 0$, as long as the flow paths picked by the routing algorithm are not broken. Thus, we can obtain a finite upper bound for this logarithm approximation.

Theorem 1: If $0 < \beta \leq D_s(t) \leq 1$ ($\forall s \in \mathcal{S}$ and $1 \leq t \leq N_t$), then

$$0 \leq \log\Big(\sum_{s\in\mathcal{S}}\sum_{t=1}^{N_t}\frac{r_s\Delta}{R_S}D_s(t)\Big) - \sum_{s\in\mathcal{S}}\sum_{t=1}^{N_t}\frac{r_s\Delta}{R_S}\log\left(D_s(t)\right) \leq \log\Big(\frac{\beta-1}{\beta^{\frac{1}{\ln\beta}+\frac{1}{1-\beta}}\ln\beta}\Big).$$

The proof is omitted due to the page limit. Theorem 1 implies that the approximation error introduced by the logarithm transform can be bounded by a constant. Converting its logarithm value back to the original objective, we have that the ratio between the optimal and approximated values is also bounded. More importantly, this approximation ratio is independent of the problem size, including the network size, flow rate and the time slot granularity.

Now we deal with the difficulty brought by the curvature of the probability distribution function $Pr(\alpha)$. For the ease of presentation, we omit α's subscripts of links and times. It can be easily verified that $Pr(\alpha)$ ($\alpha \geq 1$) is not always concave or convex, as shown in Figure 2. Interestingly, one insight which can be observed from Figure 2 is that $Pr(\alpha)$ tends to become concave as the block delivery probability p goes up. Thereby, we hypothesize that $Pr(\alpha)$ is concave when the quality of links used by flows is reasonably good. This is confirmed by the following theorem.

Theorem 2: $Pr(\alpha)$ is always concave with respect to α if the packet delivery probability

$$p_m = p^b \geq \left[\frac{b^2 - 1 + \sqrt{3(b^2-1)}}{(b+1)(b+2)}\right]^b \tag{10}$$

The proof is omitted due to the limited space. Note that $p_m = p^b$ is the successful delivery probability of a packet having a size equal to b encoded blocks, when no coding scheme is used. Although the lower bound in Theorem 2 is not tight, it is always below 29% even for very large b, as shown in Figure 3. Considering that b is usually not too large because of the extra framing overhead introduced by encoded blocks, we can see that this lower bound on the link quality can be easily satisfied by asking the routing scheme to choose links with reasonable packet delivery probability (e.g., above 28% for $b = 8$).

Observe that $\widetilde{Pr}(\alpha)$ is actually a series of line segments connecting every pair of consecutive points at $\alpha = \frac{i}{b}$ for integer $i \geq b$. Let $\ell_i(\cdot)$ be the line connecting point $(i/b, \widetilde{Pr}(\frac{i}{b}))$ and point $(\frac{i+1}{b}, \widetilde{Pr}(\frac{i+1}{b}))$. When condition (10) in Theorem 2 is satisfied, $\widetilde{Pr}(\alpha)$ is equivalent to the infimum of the set of those lines and the horizontal line $\widetilde{Pr}(\alpha) = 1$ since $\widetilde{Pr}(\alpha) \leq 1$. Formally, we have $\widetilde{Pr}(\alpha) \equiv \inf\{\ell_i(\alpha), 1 : i = b, b+1, ...\}$.

Based on the two theorems, we propose SolarCode to determine the redundancy level of erasure codes. For each link with quality p, it first computes the line functions $\ell_i(\cdot)$ used in $\widetilde{Pr}(\alpha)$. Note that $\ell_i(\cdot)$ will be close to the horizontal line $\widetilde{Pr}(\alpha) = 1$ when i is large. Thus, for computational efficiency, we compute $\ell_i(\cdot)$ only for $i = b, ..., Z$, where Z is a sufficiently large integer constant. Plugging $\widetilde{Pr}(\alpha)$ into the logarithm objective function (9), we obtain a normal convex optimization problem, and solve it for $\alpha_l(t)$, $l \in \mathcal{L} \wedge 1 \leq t \leq N_t$. The approximation ratio is bounded by Theorem 1.

IV. Performance Evaluation

In this section, we evaluate SolarCode based on the real setting of our previously published solar-powered sensor network testbed [8], [9], [10]. This outdoor testbed is on the south campus of the University of Illinois at Urbana-Champaign. Currently, 9 nodes have been deployed and running since August 2008 for environmental monitoring applications (e.g., recording bird vocalizations). Figure 4 shows the topology of our 9 deployed nodes. Packet delivery probability of links between these nodes is measured on the testbed, and only links with packet delivery probability greater than 30% are selected to form routes between source and destination nodes. Totally 6 flows are used in our evaluation. Table I shows the flow routes and data rates in packets per second (100 bytes per packet). Please refer to [8] for detailed system parameters.

ID	Rate (10^2pkt/s)	Route	ID	Rate (10^2pkt/s)	Route
f_1	1.5	2-7-1	f_4	1.25	4-5-9-3
f_2	0.5	5-7-1	f_5	2.0	3-8-1
f_3	0.5	6-2-7-9	f_6	1.0	6-4-5-9-1

TABLE I
DATA RATE AND ROUTE OF 6 FLOWS USED IN THE EVALUATION.

SolarCode relies on the projection of the available solar energy. Many analytic, stochastic, empirical or more complex (e.g., artificial neural network based) models [11] have been proposed to estimate solar radiation on the earth horizon. In this paper, we utilize an empirical model, which is based on the historical data and weather forecast information. Historical data are used to find the correlations between the solar radiation and local weather parameters. Thereby these correlations are used to project the solar radiation level based on the weather forecast information. Then the projected solar radiation level can be translated into the output current of a solar panel, given its rated power and its angle with the horizon. Real solar energy traces are also collected on the testbed. The pair of projected and real traces used in the experiments are shown in Figure 5. Besides the charging current (in Amperes) of a solar panel over 15 days, the figure also shows the total amount of solar energy collected daily in units of Ampere Hour. As we can see, the projected and real daily solar energy match very well.

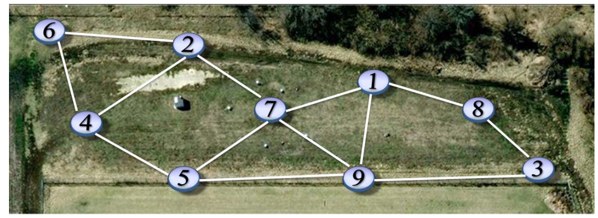

Fig. 4. Map of 9 nodes in our testbed. Only links with reasonable packet delivery probability ($\geq 30\%$) are picked in forming this topology.

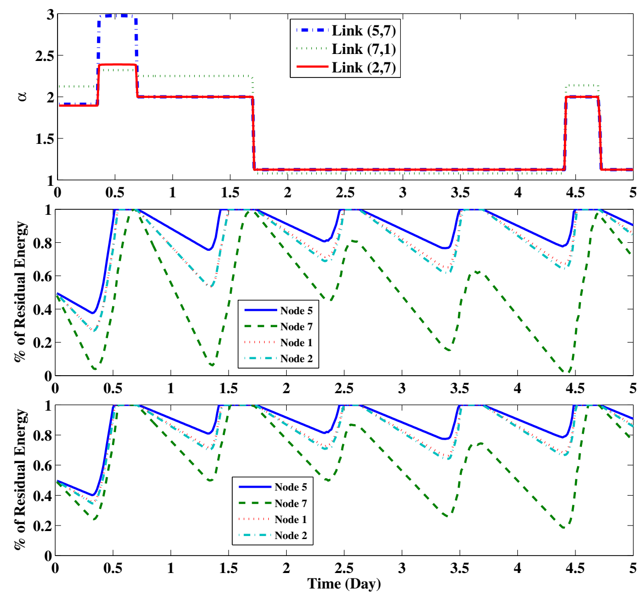

Fig. 6. The replication factor α (top) and residual energy (middle) for the experiment with flow f_1 and f_2 for 5 days. The bottom part shows the residual energy when SolarCode is not used.

We first run a basic experiment with flow f_1 and f_2 for 5 days to study how SolarCode reacts to environmental (solar energy) changes, as well as differences in link quality and traffic load. The results are plotted in Figure 6.

Four interesting observations are in order. **(1)** Replication factors α are adapted according to the energy level of sensor nodes throughout the 5 days. Normally, α is higher during daytime than during nighttime; when weather conditions are not good (e.g., rainy in day 3 and day 4), α stays at a low level to save energy for avoiding blackout. **(2)** Among the three links, link $(7, 1)$ has a higher α during the time with sufficient residual energy (day 1, 2 and 5) because it needs more redundancy than the other two links, which have better

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2010

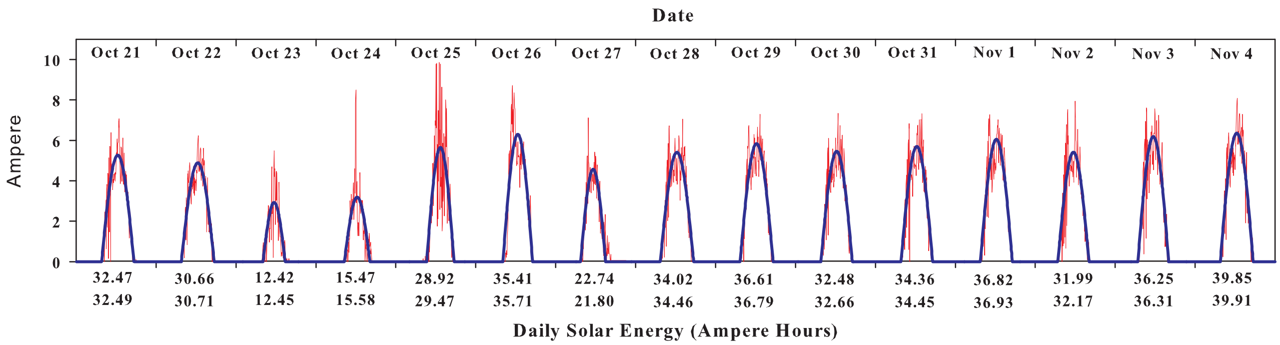

Fig. 5. Real (thin red curve) and projected (thick blue curve) solar energy traces during Oct 21st – Nov 4th 2008. Total energy collected daily in units of Ampere Hour is also shown: the numbers on the first line are for the real trace and the second are for the projected trace.

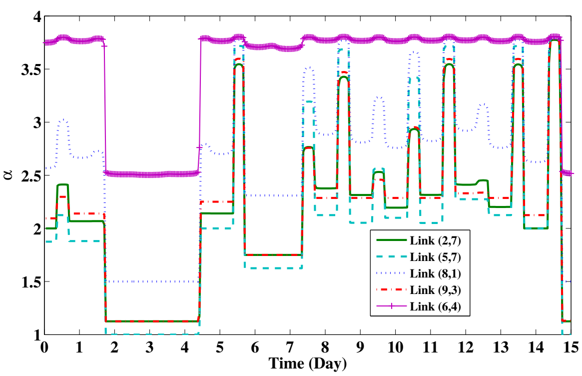

Fig. 7. The replication factor α by SolarCode for the experiment of all 6 flows for 15 days. Only 5 links are shown for the clarity of presentation.

quality than $(7, 1)$. But for days with low solar energy input, link $(7, 1)$ gets a little bit lower α than the other two links. This is because saving energy now becomes more urgent than improving reliability and more energy can be saved by lowering the α of link $(7, 1)$ due to its heavy traffic. **(3)** Link $(2, 7)$ and $(5, 7)$ have almost identical replication factors, even though the traffic carried on them is very different. The reason is that the energy consumption rate of a flow is also proportional to the flow rate. From the perspective of packets, they can reach the same reliability level by consuming the same amount of energy on the two links with the same quality. Thus there is no need to differentiate which flow these packets belong to. **(4)** As shown in the middle part of Figure 6, SolarCode only spends energy surplus (if any) to enhance the delivery reliability and hence incurs no blackout. From the bottom part of Figure 6, we can see that part of the residual energy has never been utilized when SolarCode is not used.

During the 5 days, the total number of packets delivered by the two flows is 5.56×10^7, resulting in an average end-to-end delivery probability of 63.4%, which is only 12.9% if SolarCode is not used.

In the next experiment, we test SolarCode with all 6 flows for 15 days. For clarity, only the replication factors of interesting links are shown in Figure 7. First, we observe again that replication factor αs are regularly higher during daytime except in days of bad weather (e.g., day 3, 4, 7 and 13). Second, although we argued in the basic scenario that α of a link is independent of the flow rate, this only holds when there is plenty of energy surplus on the incident sensor nodes of this link. In fact, it is not the case for most of the time. Consequently, the replication factor that a link could reach is limited by the energy availability of its incident nodes. Thus, if a node (e.g., node 5, 7 and 9) relays heavy traffic, the replication factors of its incident links (e.g., link $(2, 7)$, $(5, 7)$, $(9, 3)$) would usually be lower than those of links with low traffic (e.g. $(8, 1)$ and $(6, 4)$). Third, compared to the basic scenario, link $(2, 7)$ and $(5, 7)$ have heavier traffic and thus attain lower replication factors than in the basic scenario. For these 15 days, the average end-to-end delivery probability is 89.9% by using SolarCode, much higher than the delivery probability 19.1% when no SolarCode is used.

V. Conclusion

In this paper, we present SolarCode to dynamically adjust the redundancy level of each data link in the network, such that the end-to-end packet delivery probability is maximized and the network lifetime is not affected. Experiment results show that SolarCode schedules the redundancy level of each link dynamically according to the solar energy harvesting process.

References

[1] M. Luby, "LT codes," in *FOCS*, 2002.
[2] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 569–584, Feb 2001.
[3] J. Kangasharju, K. Ross, and D. Turner, "Optimizing file availability in peer-to-peer content distribution," in *INFOCOM*, 2007.
[4] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," in *SIGCOMM*, 2005.
[5] J. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to asynchronous reliable multicast," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 8, pp. 1528–1540, Oct 2002.
[6] Y. Wu, A. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," in *45th Annual Allerton Conference*, 2007.
[7] A. G. Dimakis, B. Godfrey, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *INFOCOM*, 2007.
[8] Y. Yang, L. Wang, D. K. Noh, H. K. Le, and T. F. Abdelzaher, "SolarStore: enhancing data reliability in solar-powered storage-centric sensor networks," in *Mobisys*, 2009.
[9] L. Wang, Y. Yang, D. K. Noh, H. Le, J. Liu, T. F. Abdelzaher, and M. Ward, "AdaptSens: an adaptive data collection and storage service for solar-powered sensor networks," in *RTSS*, 2009.
[10] D. K. Noh, L. Wang, Y. Yang, H. K. Le, and T. F. Abdelzaher, "Minimum variance energy allocation for a solar-powered sensor system," in *DCOSS*, 2009.
[11] K. S. Reddy and M. Ranjan, "Solar resource estimation using artificial neural networks and comparison with other correlation models," *Energy Conversion and Management*, vol. 44, no. 15, pp. 2519 – 2530, 2003.