On Optimal Energy Efficient Convergecasting in Unreliable Sensor Networks with Applications to Target Tracking

Srikanth Hariharan
Department of ECE
The Ohio State University
Columbus, OH-43210, USA
harihars@ece.osu.edu

Ness B. Shroff
Departments of ECE and CSE
The Ohio State University
Columbus, OH-43210, USA
shroff@ece.osu.edu

ABSTRACT

In this paper, we develop a mathematical framework for studying the problem of maximizing the "information" received at the sink in a data gathering wireless sensor network. We explicitly account for unreliable links, energy constraints, and in-network computation. The network model is that of a sensor network arranged in the form of a tree topology, where the root corresponds to the sink node, and the rest of the network detects an event and transmits data to the sink over one or more hops. This problem of sending data from multiple sources to a common sink is often referred to as the *convergecasting* problem. We develop an integer optimization based framework for this problem, which allows for tackling link unreliability using general error-recovery schemes. Even though this framework has a non-linear objective function, and cannot be relaxed to a convex programming problem, we develop a low complexity, distributed solution. The solution involves finding a Maximum Weight Increasing Independent Set (MWIIS) in rectangle graphs over each hop of the network, and can be obtained in polynomial time. Further, we apply these techniques to a target tracking problem where we optimally select sensors to track a given target such that the information obtained is maximized subject to constraints on the per-node sensing and communication energy. We validate our algorithms through numerical evaluations, and illustrate the advantages of explicitly considering link unreliability in the optimization framework.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*

General Terms

Algorithms

Keywords

Wireless sensor networks, data aggregation, target tracking, convergecasting, energy efficiency, unreliable links, integer optimization, graph theoretic algorithms

1. INTRODUCTION

In a wireless sensor network, a group of sensors are deployed to sense certain aspects of a region. The properties sensed could potentially have a number of applications in military and civilian domains, such as finding the location of a target, sensing the temperature and pressure in a region, etc. While the network is ad-hoc in general, occasionally sensors have to report the sensed data to one or more sinks (end-users). When there is only one sink, the problem of sending data from multiple sources to this common sink is often called the *convergecasting* problem [22]. In many sensor network applications, the sink only requires an aggregated form of the data sensed by different nodes (e.g., average temperature, maximum pressure, an indication of whether a target is present, etc.). In such cases, performing *in-network computation* (i.e., intermediate nodes performing a functional computation of the data that they receive from multiple sources) is known to greatly reduce the communication overhead [8]. Since sensors are typically battery operated, it is important to conserve energy so that the network can function for a long period of time without requiring much maintenance.

The energy spent by sensors is primarily because of two operations - sensing and communication; the latter involves transmissions and receptions. We are interested in maximizing the *information* that reaches the sink subject to energy constraints. For sensor networks, the *information* that reaches the sink can be a measure of the quality of the actual data that reaches the sink. For instance, it can be the sum of the inverses of the error variances of the source nodes whose data reaches the sink (this is a measure of the overall variance obtained when the measurements are independent [17]). On the other hand, if the network is detecting a signal, the information could be the sum of the Log-Likelihood Ratios obtained from the source nodes. Contrary to existing works, we also explicitly model the effects of unreliable links in our framework. Therefore, the "information" depends not only on which sensors monitor a particular event but also on the successful transmission of the data to the sink. We develop algorithms that allocate energy to each sensor node so that they can perform sensing and end-to-end communication in an efficient manner.

Target tracking is a widely studied application of sensor networks. While innumerable approaches study this problem from a signal processing perspective, very few works study this problem considering multi-hop wireless networks, unreliable links, and communication energy. As a target moves through the network, the set of sensors that track the target also need to dynamically change because there may exist better sensors to take a measurement during the next sampling instant compared to the current ones. Also, there may exist sensors that have large amounts of energy left over compared to others. Therefore, in order to maximize the information that the sink obtains considering these energy constraints, we need to select sensors appropriately.

All these optimization problems are naturally integer programming problems. For instance, selecting a sensor can be modeled by an indicator variable, and the communication energy spent depends on the number of transmissions and receptions. Although many integer programming problems are NP-Hard, we show that the problems that we consider here can be efficiently solved by low-complexity, distributed algorithms.

The main contributions of this work are as follows.

- We develop an integer programming based optimization framework for solving the problem of *maximizing the information for sensor networks with unreliable links under energy constraints, considering non-linear, and potentially non-concave objective functions*. This framework explicitly accounts for general error recovery schemes when links are unreliable, as long as errors across links are independent.

- We prove that the optimal solution is obtained by finding a *Maximum Weight Increasing Independent Set (MWIIS) in rectangle graphs*, which can be found in *polynomial time*. We apply this technique to develop a low-complexity, distributed optimal solution to our problem.

- We apply our framework to a target tracking problem, and provide an algorithm that *selects sensors to activate* for sensing and communicating information about a target during each sampling period. This algorithm *maximizes the information received by the sink while explicitly accounting for unreliable links, and per-node energy constraints.*

- We provide extensive numerical validations that illustrate the importance of including unreliability in links in the problem framework.

The rest of this paper is organized as follows. In Section 2, we overview related work. In Section 3, we describe our system model and assumptions. In Section 4, we discuss graph theoretic fundamentals that are of importance to this paper. In Section 5, we study our energy constrained problem for known source nodes, and provide a distributed optimal solution. In Section 6, we apply our techniques to select sensors for tracking targets in an energy efficient manner. In Section 7, we provide numerical results that illustrate the importance of considering link unreliability in the problem framework. Finally, in Section 8, we conclude the paper.

2. RELATED WORK

A number of interesting problems arise in convergecasting in wireless networks. In [2], [4], the authors have considered the problem of minimizing the number of time slots required to send all the packets in a wireless network to a sink under different interference constraints. While these works do not consider in-network computation, in [7], the authors have studied this problem for a one-hop interference model with deadline constraints and in-network computation. Zhang et. al., [22] study reliable convergecasting in wireless networks. They provide heuristic algorithms to increase end-to-end packet delivery ratio and decrease delay.

Tree structures are optimal data gathering structures in a number of cases. However, constructing an optimal tree is an NP-Hard problem in many of these cases. Constructing efficient data gathering trees has been studied in [19] for maximizing network lifetime, in [6] for minimizing the expected cost of the tree, and in [13] for minimizing the total number of transmissions.

There also exist a number of works studying the trade-offs between energy, latency, and data quality. Yu et. al., [21] study trade-offs between energy and latency in data gathering trees. In [20], Ye et. al. use a decision process approach to study energy-delay trade-offs. In [14], the authors attempt to balance the communication load on all nodes using network flow optimization techniques. None of these works explicitly consider the effect of unreliable links in their framework.

The sensor activation problem for target tracking is also a well studied problem from a signal processing perspective. For example, [3] studies the problem of minimizing the total energy of activating sensors such that the error variance (computed according to an Extended Kalman Filter) is less than a threshold. This problem is computationally hard to solve, and the authors propose heuristic integer programming algorithms. Joshi et. al., [12] investigate the problem of minimizing the error variance of the estimate (for a set of linear measurements) such that the number of sensors activated is constrained. They solve this problem using convex relaxation techniques. In [18], Williams et. al., provide a dynamic programming approach that integrates the information obtained and the associated communication cost. These works provide interesting techniques when fusing information from multiple sensors results in an overall information that is not a sum of the information from individual sensors. However, these works do not consider unreliable links in their framework. While [3] and [12] do not even consider multi-hop transmissions, the communication model in [18] could result in certain nodes transmitting many more times than others.

3. SYSTEM MODEL

We model the system as a graph $G(V \cup \{S\}, E)$ where V is the set of *active* nodes, S is the sink, and E is the set of links. The graph G is a tree rooted at the sink. An *active* node is one that is either a source node or has at least one source node in its sub-tree. Except for the target tracking problem considered in this paper, we assume that source nodes are known. When an event occurs, nodes sense some desired quantity, and send an aggregated form of the data to the sink. A node may or may not be a source for a particular event. For notational convenience, we consider only one event for this problem. However, it is straightforward to extend it to multiple events when the total information is expressed as a weighted sum of the information from each event.

We refer to the actual sensor measurements as *data*. We define *information* as the quality of the data. For example, data could be temperature, location, etc., while information could be error variance, distortion, Log-Likelihood Ratio, etc. Let w_i represent the information provided by a source node i. w_i, for example, could represent the inverse of the predicted variance of a Kalman filter if node i is tracking a target.

Every node can perform in-network computation of the data that it receives. We assume "perfect" aggregation, i.e., nodes can transmit aggregated data in a single transmission. Even a single transmission could consume a large amount of energy depending on the size of the data. For example, the data could be a high-resolution image. Therefore, whenever it is possible to combine data from multiple sensors in a meaningful manner, one should do so in order to save energy [8]. This model of aggregating and transmitting is also suitable for function computation. In particular, we can use this setup if the sink requires any *divisible function* [5] of the sensor measurements. Divisible functions are those that can be computed in a divide and conquer fashion. The complete definition can be found in [5]. Examples of divisible functions include MIN, MAX, Sum, Mean, Higher Order Statistics, etc. In this paper, we do not consider "imperfect" aggregation, i.e., the data aggregated from multiple sensors requires a certain number of transmissions that is a function of the number of sensors, and the type of data that is aggregated. While our model can be extended to allow for "imperfect" aggregation, the computational complexity of the solution significantly increases. It is beyond the scope of this paper to discuss elaborately on these issues, and hence we leave it for future work. We note that the notion of "perfect" aggregation has been used in a number of works in the literature addressing data aggregation in sensor networks [6–8, 10, 13, 19, 20].

We assume per-node energy constraints, where the energy spent by a node is the sum of the transmission and reception energy expended at that node. We allow for unreliable links in the network. We assume that errors are independent across links. Since links are unreliable, we can employ a number of known error-recovery techniques such as retransmissions, coding, etc. We assume a function $f_i(t)$ for each link i, where $f_i(t)$ is the *link reliability* of link i if t units of energy is expended on transmissions over link i. For instance, if we use retransmissions, $f_i(t)$ could denote the probability that a packet is successfully transmitted over link i if a maximum of t transmissions (including retransmissions) are allowed. From a practical point of view, $f_i(t)$ must be an increasing function of t. Since there are no other restrictions on $f_i(\cdot)$, it can model any practical error recovery scheme for known channel conditions. We consider two definitions of the *information received at node B from a particular source node A*.

Definition 1: Product of the information provided by node A and the *product (or weighted product)* of the *link reliabilities* over all the links from node A to node B. For example, if $P_{A,B}$ represents the path from A to B, then the information received at B from A is $w_A \prod_{j \in P_{A,B}} f_j(t_j)$.

Definition 2: Product of the information provided by node A and the *sum (or weighted sum)* of the *link reliabilities* over all the links from node A to node B. For example, the information received at B from A (over a path $P_{A,B}$ from A to B) is $w_A(\sum_{j \in P_{A,B}} f_j(t_j))$.

We model the *information received at node B* as a (weighted) sum of the information received at B from individual source nodes, where the latter is obtained from Definition 1 or Definition 2. Note that when B is the sink, we obtain the information received at the sink.

Both the definitions have practical implications. While Definition 1 could represent the probability that a packet from the source reaches the sink if errors are independent across links, Definition 2 could represent the utility gained by using a particular link. Further, the overall information is also of importance in practice for performing fusion (see Section 3.1). Note that expressing the overall information as a sum of individual sensor information has been assumed in a number of works in the data aggregation literature [7, 9, 10, 13, 16, 20]. While these metrics of information model a large class of practical information metrics, they may not be suitable if the information gathered from multiple sensors is not a weighted sum of the individual sensor information. For example, the information gathered from a magnetic sensor, and a video sensor may not be a weighted sum of the individual sensor information. When the overall information is an arbitrary function of the information from individual sensors, the function may no longer be separable, and hence the problem becomes NP-Hard. One may have to individually deal with each such non-separable function (as done in [3], [12] for error-free links, and single-hop networks). Extensions to arbitrary functions for the overall information corresponding to specific sensor network applications is therefore another open problem for future research. Our focus in this paper is to deal with the large class of cases when our metric is relevant and to consider energy allocation in a multi-hop sensor network with unreliable links.

It is important to note here that while the overall information received at the sink is the sum of individual sensor information, the overall aggregated data packet received is just an aggregated version of raw data measurements of sensors. For instance, if the sink desires the maximum temperature in a region, the aggregated data packet just contains the maximum temperature. The information provided by this aggregated data packet is the sum of the information (e.g., the inverse of the measurement error variance) provided by individual temperature sensors in the network.

Since our definitions are abstract, we now provide an application of our model in wireless sensor networks.

3.1 Motivating application - sensor networks

Consider a sensor network shown in Figure 1. Sensors send their data over a tree network to a sink (red node). In order to illustrate our definitions of information, consider a part of this network (encircled in green) with three nodes and the sink S. Suppose that all the three sensors are source nodes measuring the location of a target. Let sensor i provide an error variance R_i for the target location, $i \in \{1, 2, 3\}$. Then, a metric for measuring the *information* provided by sensor i is $\frac{1}{R_i}$. If data from sensors 1 and 2 is combined (say), then one of the metrics for the overall variance, R_{12} of the combined data is given by $\frac{1}{R_{12}} = \frac{1}{R_1} + \frac{1}{R_2}$, which is lower than both R_1 and R_2 [17].

Suppose that retransmissions are used in order to account for unreliable links. If sensor i is allowed to make t_i transmissions, the probability that sensor i's transmission is successful is given by $f_i(t_i)$. When data aggregation is used at intermediate nodes, for example, node 1 can

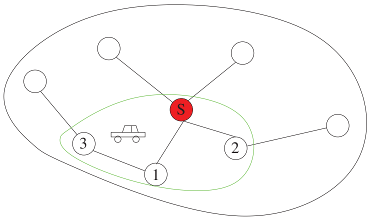


Figure 1: Motivating Example

combine data from node 3, and transmit both data simultaneously. Assuming that errors across links are independent, the probability that node 1's data reaches the sink is $f_1(t_1)$, the probability that node 2's data reaches the sink is $f_2(t_2)$, and the probability that node 3's data reaches the sink is $f_3(t_3)f_1(t_1)$. Now, the expected information received at the sink is $\sum_{i\in\{1,2,3\}} \frac{1}{R_i}\text{P}(i\text{'s data reaches the sink}) = \frac{1}{R_1}f_1(t_1)+\frac{1}{R_2}f_2(t_2)+\frac{1}{R_3}f_3(t_3)f_1(t_1)$. One can clearly verify that without taking the communication model into account, we get the information $\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}$. Note that this metric of information for each sensor corresponds to Definition 1.

Definition 2 is also of importance in practice. Suppose that we desire proportional fairness [15] when allocating number of transmissions for nodes. This can be represented by the function log(P(i's packet reaches the sink)). Now, for sensor 3, $\log(f_3(t_3)f_1(t_1))$ becomes $\log(f_3(t_3))+\log(f_1(t_1))$, which corresponds to Definition 2.

Table 1: Notations and Definitions

V	Set of *active* sensor nodes.
S	Sink.
V_S	Set of all source nodes.
V_L	Set of all leaf nodes in the tree.
E	Set of edges.
$P(i)$	Parent of node i.
$C(i)$	Set of children of node i.
t_i	The number of transmissions made from i to its parent $P(i)$.
e_i	The energy constraint on node i.

4. PRELIMINARIES

We use the following graph-theoretic notions.

1. **Independent Set (IS)**: An Independent Set is a set of vertices in a graph, no two of which have an edge between them.

2. **Maximum Weight Independent Set (MWIS)**: An IS for a given graph such that the total weight of the vertices is maximum over all ISs in that graph.

3. **Interval Graphs**: Let $\{I_1, I_2, ..., I_n\}$ be a set of intervals on the real line. Then, the interval graph $G(V, E)$ corresponding to this set of intervals is defined as follows.

a. $V = \{I_1, I_2, ..., I_n\}$. Each vertex is an interval.

b. For any $y, z \in \{1, 2, .., , n\}$, $(I_y, I_z) \in E$ if and only if the intervals intersect, i.e., $I_y \cap I_z \neq \emptyset$.

4. **Interval graph of interval number m:** The definition is identical to that of the interval graph except that each vertex can now be represented as a disjoint union of m intervals. Two vertices will have an edge between them if and only if at least one of the intervals corresponding to one vertex have a non-empty intersection with at least one of the intervals corresponding to the other vertex. An interval graph of interval number 2 is called a *double interval graph.*

5. **Rectangle graphs:** Rectangle graphs are a subclass of double interval graphs. A double interval graph can be transformed into a rectangle graph by simply labeling the vertices in the double interval graph as the set-product of the two intervals instead of the union of the two intervals. Thus, each vertex now represents a rectangle in $\mathbb{R}^2$. Note that two rectangles that do not overlap need not form an IS in the corresponding double interval graph. On the other hand, every IS in the double interval graph is an IS in the rectangle graph.

6. An MWIS on interval graphs (order 1) can be found in polynomial time. However, MWIS on interval graphs of order m, $m > 1$, is still NP-Hard [1].

7. **Increasing Independent Set (IIS) on rectangle graphs:** An Increasing Independent Set (IIS) on a rectangle graph is an IS that has the following property. Let $Z = \{r_1, r_2, ..., r_m\}$ be an ordered set of rectangles ordered in the following fashion. For any $i, j \in \{1, 2, ..., m\}$ such that $i < j$,

a. The maximum x-coordinate of any point in $r_i \leq$ the minimum x-coordinate of any point in r_j.

b. The maximum y-coordinate of any point in $r_i \leq$ than the minimum y-coordinate of any point in r_j.

Then, Z is an IIS on the given rectangle graph. The rectangles in Z are ordered such that the next rectangle is *to the right and to the top* of the previous rectangle in the order. Further, an IIS on a rectangle graph is an IS on the corresponding double interval graph.

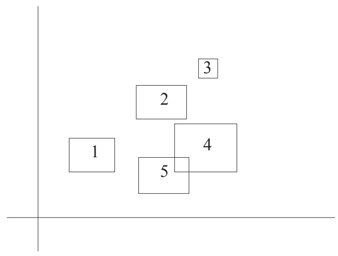


Figure 2: Rectangle Graphs and Independent Sets

The following example illustrates the definitions above (Figure 2). Each rectangle in the figure represents a vertex in a rectangle graph. There will be an edge between two vertices only if the corresponding two rectangles intersect. For instance, there will be an edge between rectangles 4 and 5. $\{1, 2, 3, 4\}$ and $\{1, 2, 3, 5\}$ are *two maximal ISs of rectangles.* While $\{1, 4\}$ is an IS in the rectangle graph, it does not form an IS in the corresponding double interval graph because its intervals on the y-axis intersect. Also, $\{1, 2, 3\}$ is an example of an IIS in the rectangle graph because rectangle 2 is to the right and to the top of rectangle 1, and rectangle 3 is to the right and to the top of rectangle 2.

5. MAXIMIZING INFORMATION - KNOWN SOURCES

In this section, we formulate and solve the problem of maximizing information in sensor networks under per-node energy constraints. Let λ_i represent whether a node is a source for a particular event, i.e., $\lambda_i = 1$, if i is a source, and $\lambda_i = 0$, otherwise.

Problem Π_E:

$$\max_{\vec{e}} \sum_{i \in V_S} w_i(\text{Information received at } S \text{ from } i \text{ (Defn.1)})$$

$$\text{s. t.} \quad (1)\ t_i T \leq e_i, i \in V_L,\ (2) \sum_{j \in C(S)} t_j R \leq e_S$$

$$(3)\ t_i T + \sum_{j \in C(i)} t_j R \leq e_i, \text{ for other nodes}$$

$$(4)\ t_i \in \{0, 1, ..., \beta_i\}$$

In Problem Π_E, w_i is a weight for source node i, T is the energy spent for one transmission, and R is the energy spent for one reception. For notational convenience, we have assumed that T and R are the same for all nodes in the network. It is straightforward to extend this to the case where each node has its own T and R. Also, while we have mentioned that the information received at S from i is obtained according to Definition 1, we can use Definition 2 as well. The solution methodologies are identical for both definitions, and WLOG, we use Definition 1 from now on. The first constraint is for leaf nodes in the tree. These nodes do not make any receptions. The second constraint is for the sink. The sink does not make any transmissions. The third constraint represents the energy constraint for an intermediate node in the tree.

Finally, for each $j \in C(i)$, $t_j \in \{0, 1, ..., \beta_j\}$, where $\beta_j = \min(r, \lfloor \frac{e_j}{T} \rfloor)$. β_j represents the maximum amount of energy that can be spent over a link to achieve a certain reliability. For example, if retransmissions are used, β_j could represent the maximum number of retransmissions that can be allowed over a link. β_j depends on the link characteristics. For instance, if retransmissions are used, and the probability of error of a link is 0.1, and we require a success rate of at least 0.99, β_j can be obtained by solving $0.1^{\beta_j} \leq (1 - 0.99)$. This gives $\beta_j \geq 2$.

We have used the following forwarding policy while formulating Π_E. Due to this policy, each link $(i, P(i))$ is assigned only one energy variable t_i. This means that the node aggregates packets from its sub-tree, and uses this energy to transmit the aggregated packet.

Forwarding Policy: For an event, each node will wait to gather data from its predecessors. It will perform in-network computation on the data that it receives, and transmit the aggregated data. The implication of this policy is that a node will never transmit data from an event in separate transmissions.

We now show that this forwarding policy is optimal.

THEOREM 5.1. *Consider an optimization problem Π'_E for maximizing the information in a data aggregation tree under per-node energy constraints. Suppose that Π'_E is not restricted to a particular forwarding policy. Any optimal solution to Π_E is also an optimal solution to Π'_E.*

PROOF. We prove the result by contradiction. Suppose that an optimal solution to Π_E is not optimal to Π'_E. This means that according to Π'_E, there exists a node i that transmits data from at least two different sources (say, 1 and 2) in separate transmissions. Suppose that the optimal solution to Π'_E allocated energy $t_i^{(1)}$ for source 1, and energy $t_i^{(2)}$ for source 2. Then, the link reliabilities over the link from i to its parent for these sources are $f_i(t_i^{(1)})$ and $f_i(t_i^{(2)})$, respectively. If i had aggregated data from sources 1 and 2, then it could have allocated an energy $t_i^{(1)} + t_i^{(2)}$ for the aggregated data. Clearly, $t_i^{(1)} + t_i^{(2)}$ is a feasible energy allocation since both $t_i^{(1)}$ and $t_i^{(2)}$ are feasible energy allocations for node i. Therefore, if i had aggregated data from sources 1 and 2, the information received by i's parent would have been $f_i(t_i^{(1)} + t_i^{(2)})$ for both the sources. Since $f(\cdot)$ is an increasing function, $f_i(t_i^{(1)} + t_i^{(2)}) \geq f_i(t_i^{(1)})$, and $f_i(t_i^{(1)} + t_i^{(2)}) \geq f_i(t_i^{(2)})$. Therefore, by gathering data, i would have accounted for at least as much information as otherwise.

This contradicts our assumption that an optimal solution to Π_E is not optimal to Π'_E. Hence, the forwarding policy is optimal. $\square$

We now develop a distributed optimal solution to Π_E. We do this by rewriting the objective function in a recursive manner. Let $X[i, r]$ represent the maximum information that can be received at node i from its sub-tree if it allows r units of energy for reception. We wish to determine $X[S, \lfloor \frac{e_S}{R} \rfloor]$.

For $i \in V_L$ (i is a leaf node), $X[i, r]$ can be calculated as $X[i, r] = w_i \lambda_i$.

For any non-leaf node, and for a given r, $X[i, r]$ can be represented in the following recursive manner.

$$X[i, r] = w_i \lambda_i + \max_{\sum_{j \in C(i)} t_j \leq r} \sum_{j \in C(i)} X[j, \lfloor \frac{e_j - t_j T}{R} \rfloor] f_j(t_j) \tag{1}$$

If Definition 2 is used in the objective function instead of Definition 1, $X[j, \lfloor \frac{e_j - t_j T}{R} \rfloor] f_j(t_j)$ will be replaced by the sum $(X[j, \lfloor \frac{e_j - t_j T}{R} \rfloor] + f_j(t_j))$.

Theorem 5.2 shows the correctness of the recursion. Before we describe the details, we provide the intuition behind this recursion. $X[i, r]$ represents the maximum information received at node i from the sub-tree rooted at node i if i allows r units of energy for reception. This information is the sum of i's own information, given by $w_i \lambda_i$, and the information received from i's children. Since $f_j(t_j)$ represents the link reliability of the link from j to i, according to Definition 1, the information provided from j to i is the product of the information it has and the link reliability of the link (j, i). By choosing transmission and reception energies at children carefully, one can maximize the information at i while satisfying the constraint that i can allow at most r units of energy for reception.

Further, it can be seen that as the amount of reception energy, r, is varied, the amount of data that i's children can transmit to i, and hence the information that reaches i also varies. As r increases, the energy allowed for reception at i increases, and hence the information provided by i's children increases. Thus, $X[i, r]$ increases. However, as r increases, the amount of energy that i can allocate for transmission (to its parent) decreases due to the per-node energy constraint on i. Therefore, i may not have sufficient energy left for transmissions if r is too high. The rest of the paper focuses on how to choose transmission and reception energies at each node optimally such that the information received at the sink is maximized.

THEOREM 5.2. *For any node i, and for any given r, $X[i, r]$ (Equation (1)) maximizes the information received at node i if i allows r units of energy for reception.*

PROOF. We prove this result by induction on the distance (number of hops) of the node from the root.

It is straightforward to see that for a leaf node, Equation (1) reduces to $X[i, r] = w_i\lambda_i$. Since a leaf node does not have any children, this is the maximum information that reaches a leaf node i for any value of r.

Assume that the result is true for all nodes that are h hops from the root.

Consider a node i which is $h-1$ hops from the root. If i allocates r units of energy for reception, then the transmissions of i's children is clearly constrained by the relation, $\sum_{j\in C(i)} t_j \leq r$. For each child $j \in C(i)$, if j allocates t_j units of energy for transmission, then due to the per-node energy constraint on j, it can at most allocate $\lfloor\frac{e_j - t_jT}{R}\rfloor$ units of energy for reception. By the induction hypothesis, $X[j, \lfloor\frac{e_j-t_jT}{R}\rfloor]$ provides the maximum information that j can account for if it allows $\lfloor\frac{e_j-t_jT}{R}\rfloor$ units of energy for reception. According to Definition 1, the information that reaches i from j is given by $X[j, \lfloor\frac{e_j-t_jT}{R}\rfloor]f_j(t_j)$. Since the information received at i is the weighted sum of the individual information (and the weights are accounted for at the source nodes), by solving $\max_{\sum_{j\in C(i)} t_j \leq r} \sum_{j\in C(i)} X[j, \lfloor\frac{e_j - t_jT}{R}\rfloor]f_j(t_j)$, we can obtain the maximum information at node i. Thus, $X[i, r]$ (Equation (1)) maximizes the information that reaches node i if i allows r units of energy for reception. □

The interesting and difficult part in solving problem Π_E is to solve $X[i, r]$. We develop the solution as follows.

LEMMA 5.3. *$X[i, r]$ can be obtained by finding an MWIS in the graph G' defined as follows.*

- *Create nodes labeled (j, I_j, t_j) for each $j \in C(i)$, $I_j \in \{\max(0, r-\sum_{j\in C(i)} \beta_j), \max(0, r-\sum_{j\in C(i)} \beta_j)+1, ..., r-1\}$, and $t_j \in \{0, 1, ..., \beta_j\}$. The first term represents the child, and the second and the third terms represent the interval $\{I_j, I_j+1, ..., I_j+t_j-1\}$. The length of this interval, t_j, is the transmission energy allocated to child j.*
- *Assign a weight $X[j, \lfloor\frac{e_j-t_jT}{R}\rfloor]f_j(t_j)$, for a node labeled (j, I_j, t_j).*
- *Consider any two nodes (m, I_m, t_m) and (n, I_n, t_n) in G'. Create an edge between these two nodes either if they represent the same child, or if the corresponding intervals intersect, i.e., either if $m = n$, or if $\{I_m, I_m + 1, ..., I_m + t_m - 1\} \cap \{I_n, I_n + 1, ..., I_n + t_n - 1\} \neq \emptyset$.*

PROOF. From the construction of G', an IS in G' has the following properties: (a) A child can have at most one interval assigned to it. This is because of the fact that there exists an edge between any two nodes that represent the same child; (b) Two children cannot be assigned intervals that have a non-empty intersection. This also follows from the construction of edges in G'. Also, by construction, the sum of the lengths of the intervals of all the nodes in any IS in G' cannot exceed r. Therefore, the sum of the energy received by i is less than rR. Thus, every IS in G' satisfies the reception energy constraint of node i.

Further, for each possible allocation of energy to the children that satisfy the reception energy constraint of i, there exists an IS in G' corresponding to that allocation. This can be proven as follows. Suppose that $|C(i)| = k$, the energy allocation of child j is t_j, and we have $\sum_{j\in C(i)} t_j \leq r$. Then, an IS corresponding to this energy allocation is $\{(C_1, I_1, t_1), ..., (C_k, I_k, t_k)\}$, where $I_1 = r - \sum_{j\in C(i)} t_j$, $I_2 = I_1 + t_1$, ..., $I_k = I_{k-1} + t_{k-1}$. Clearly, $I_m \cap I_n = \emptyset$ for $m, n \in \{1, 2, ..., k\}$ and $m \neq n$. Thus, the constraint that the energy allocated must be in intervals that do not intersect is equivalent to the constraint that the sum of the received energy must be at most r, and hence we can find $X[i, r]$ by finding an MWIS in G'. □

We now provide an example explaining the construction of G'. Figure 3 illustrates a hop with two nodes, A and B, having parent P. Assume that $\beta_A = 2$, and $\beta_B = 1$. For simplicity of illustration, assume that each transmission and reception cost one unit of energy. Then, for a reception energy constraint r for P, the graph G' is shown in Figure 4. The boxes in this figure represent cliques, i.e., there is an edge between each node in a box and every other node in the box. Edges between nodes in two different boxes are as shown in the figure. The reason that we have these cliques is that a node having label A as the first term conflicts with all other nodes having A as their first term. The same holds for B. Further, if the energy allocation intervals corresponding to A and B have a non-empty intersection, then there are edges between these intervals in G'.

Suppose that $r = 3$. $(A, 1, 2)$ and $(B, 0, 1)$ form an IS. When the total reception energy at P is constrained to be at most 3, one of the solutions is to allocate 2 units of energy to A and 1 unit of energy to B. Thus, we see that an IS satisfies the total reception energy constraint. Consider another example. We can observe that $(A, 0, 2)$ and $(B, 0, 1)$ do not form an IS because they share the same first slot. However, we can see that this can be used as a solution to the energy problem since A is still allocated 2 units of energy and B is allocated 1 unit of energy. Moreover, this solution achieves the same amount of information at P as the IS solution. While a set that is not an IS could also sometimes satisfy the energy constraint, using this IS structure allows us to develop an efficient algorithm for determining the optimal solution to problem Π_E.

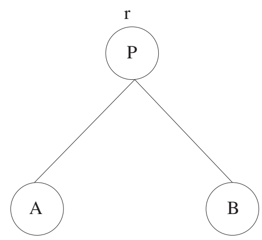


Figure 3: Two children

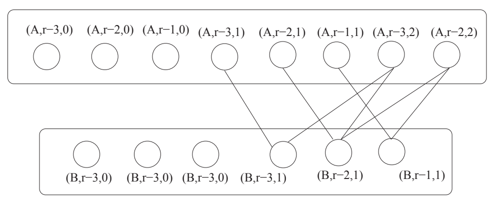


Figure 4: Graph G' (Boxes represent cliques)

Finding an MWIS is an NP-Hard problem in general graphs.

However, we now show that we can find an MWIS in G' in polynomial time.

Lemma 5.4. *The order in which i's children are allocated energy does not affect the computation of $X[i, r]$.*

Proof. The proof follows from the fact that the weight of a node (j, I_j, t_j) in G' only depends on j and t_j, and not on I_j. Since I_j represents the beginning of the interval assigned to child j, this means that the weight of the node (j, I_j, t_j) depends only on the length of the interval, and not the end-points of the interval. Therefore, the order in which children are allocated energy in a hop does not affect the computation of $X[i, r]$. □

Theorem 5.5. *Allocate intervals $[a_i, b_i]$ to $j \in C(i)$ such that $r < a_1 < b_1 < a_2 < b_2 < ... < a_k < b_k$, where $k = |C(i)|$. We then have the following results.*

1. *G' is a double interval graph. Further, by representing each node as a cross-product of two intervals instead of a union, G' becomes a rectangle graph.*
2. *$X[i, r]$ can be obtained by finding an MWIIS in this rectangle graph.*

Proof. From Lemma 5.4, we know that the order in which children are allocated energy does not affect the computation of $X[i, r]$. Therefore, if $C(i) = \{C_1, ..., C_k\}$, WLOG allocate energy starting from C_1 to C_k. This means that the energy interval allocated to a node C_j is to the left of the energy intervals allocated to all nodes C_l such that $l > j$. A node (C_j, I_j, t_j) in G' can now be represented as a union of two disjoint intervals, $[a_j, b_j]$ and $[I_j, I_j + t_j - 1]$ (since $r < a_1$, these intervals are disjoint.) Thus, G' is a double interval graph, and by representing each node as a cross-product of two intervals instead of a union of two intervals, G' becomes a rectangle graph.

By Lemma 5.3, finding an MWIS in G' provides $X[i, r]$. An IIS in the rectangle graph corresponding to G' will have the following property. For any two children C_m and C_n such that $m < n$, the rectangle corresponding to the energy allocated to C_m will be to the bottom and to the left of the rectangle corresponding to the energy allocated to C_n. This is because $b_m < a_n$, and C_m is allocated an energy interval that is to the left of the energy interval allocated to C_n in the double-interval graph. Therefore, by finding an IIS of maximum weight in this rectangle graph, we can obtain an IS of maximum weight in the double-interval graph. Hence, $X[i, r]$ can be obtained by this procedure. An MWIIS in a rectangle graph can be found in polynomial time [11]. Hence, we can find $X[i, r]$ in polynomial time. □

Figure 5 shows an IIS for the example illustrated in Figure 3. An IIS is given by $A \times \{r - 3, r - 2\}$, and $B \times \{r - 1\}$. Clearly, the rectangle for B is to the top and to the right of the rectangle for A.

Algorithm for finding an MWIIS on rectangle graphs: In [11], the authors have studied the problem of modeling similarities among DNA sequences, and have shown that their problem can be solved by finding an MWIIS on rectangle graphs constructed appropriately. This is an entirely different problem, and it turns out that both our problem, and this problem of modeling DNA sequence similarities can be approached in this manner. The authors have also provided an algorithm having a complexity $O(n \log n)$ for finding an MWIIS.

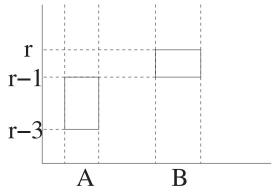


Figure 5: Increasing Independent Set

As of now, we have only provided an approach for calculating $X[i, r]$ for a particular i and r. We now provide an algorithm that calculates $X[S, \lfloor \frac{e_S}{R} \rfloor]$.

Table 2: Algorithm 1

1	Start from the leaves. For a leaf node l, for each r, $X[l, r] = w_l \lambda_l$.
2	Consider a non-leaf node i such that $X[\cdot, \cdot]$ has been determined for all children of i. For each r, calculate $X[i, r]$ by finding an MWIIS in the rectangle graph as shown in Theorem 5.5.
3	Finally, determine $X[S, \lfloor \frac{e_S}{R} \rfloor]$ at the sink.
4	Look up $X[S, \lfloor \frac{e_S}{R} \rfloor]$ to optimally allocate transmission and reception energy to the sink's children.
5	Proceed down to the leaves allocating transmission/reception energy to nodes in each hop from the root to the leaves.

Theorem 5.6. *Algorithm 1 provides an optimal solution to the problem Π_E.*

Proof. From Theorem 5.5, we know that for any node i, $X[i, r]$ provides the maximum information received at node i by the sub-tree rooted at i if i allows r units of energy for reception. Thus, $X[S, \lfloor \frac{e_S}{R} \rfloor]$ provides the maximum information received at the sink S by the tree rooted at S. Since $X[S, \lfloor \frac{e_S}{R} \rfloor]$ is the optimal value, it is straightforward to see that by looking up $X[\cdot, \cdot]$, we can obtain optimal energy allocations at each node. □

We now study the computational complexity of Algorithm 1. Consider a tree having a maximum of h hops and, a maximum of k children in each hop. Suppose that r_{max} is the maximum number of times that $X[i, \cdot]$ needs to be computed for any node i in the network. r_{max} represents the maximum number of values of the reception energy that any node in the network can possibly take. Graph G' has $O(k^2)$ nodes because there are k children and each child has $O(\sum_{j=1}^{k} \beta_j) = O(k)$ nodes. Each node in G' is mapped to a rectangle with a certain weight. Thus, constructing the set of rectangles requires $O(k^2)$ complexity. The complexity of finding an MWIIS in this graph is $O(k^2 \log(k))$. Therefore, the computational complexity of Algorithm 1 is $O(hk^2 \log(k) r_{max})$ assuming that nodes at the same level of the tree can perform computations in parallel. The computational complexity at any node in the network is $O(k^2 \log(k) r_{max})$ which just depends on the number of children and r_{max}. Typically, r_{max} is a constant. We can see that Algorithm 1 has a very low complexity, and is distributed.

6. APPLICATIONS TO TARGET TRACKING

We now discuss an application of our framework for "dynamic events" such as target tracking. When the event is dynamic, in the sense that it traverses through the network during each sampling period, the nodes that sense the event need to change dynamically, and hence we cannot assume that source nodes are known (as assumed in Π_E). Therefore, not only do we need to consider unreliable links and communication energy, but also the problem of selecting source nodes appropriately, and the sensing energy that they consume. We again consider a tree with the root as the cluster head. Motivated by the fact that many filters including Kalman filters can predict the information obtained (predicted location, predicted variance) by a future measurement, we model a framework to select sensors during the next sampling instant such that the predicted information that reaches the sink is maximized. By repeating our model for each measurement instant after updating the predicted information, we can track targets in an energy-efficient manner over a period of time.

For a given mobility model for the target, let α_i represent the information that sensor i can provide during the next sampling instant, if sensor i is selected to track target i, and sensor i successfully communicates its measurement (over multiple hops) to the root of the tree. There is a per-node energy constraint of e_i at each node i where the energy expended at node i comprises the sensing energy, and the transmission and reception energy. Let a_i be the indicator variable representing whether sensor i is chosen to track the target during the next sampling instant, i.e., $a_i = 1$, if sensor i is selected, and 0, otherwise.

Let $f_i(n)$ be the probability that i's transmission is successful if n units of energy is allocated for transmissions. Under this setup, the *expected amount of predicted information* that node i can provide during the next sampling instant is given by $\alpha_i a_i P(i$ successfully sends its packet to the root), where $P(i$ successfully sends its packet to the root) is given by Definition 1. Here, the *expectation* is related to link unreliability, and the *prediction* is related to the target's mobility model.

We wish to maximize the sum of the expected amount of predicted information over all the nodes in the network. This metric, for example, could maximize the sum of the Fisher information over all the nodes in the network when the Fisher information can be represented as a scalar value [17]. For example, consider a target moving according to a linear mobility model given by $x(t+1) = \gamma x(t) + v(t)$, where t represents the time-step, x represents the location of the target, and v is AWGN with variance Q, and zero mean. For simplicity, assume that x is a scalar. Then, if a linear Kalman filter is used for obtaining the predicted variance at time $t+1$, this variance is given by $\gamma^2 P(t) + Q$, where $P(t)$ is the variance obtained at time t. The corresponding information can be represented as $\frac{1}{\gamma^2 P(t)+Q}$. Note that this is only an example, and we can accommodate a general mobility model, and a general filtering technique to obtain the predicted information α_i for a sensor i, as long as α_i is a scalar.

Problem Π_T is shown below. M is the energy used for sensing. As explained for Problem Π_E, the first constraint represents the energy constraint for leaf nodes (no receptions). The second constraint represents the energy constraint for the sink (no sensing and transmissions). The third constraint represents the energy constraint for an intermediate node in the tree that can perform sensing, transmissions, and receptions. As in Π_E, for notational convenience, we assume that sensors use the same amount of energy M for sensing, R for receptions, and T for transmissions. In general, each node can have its own sensing and communication energy, i.e., the sensors need not be homogeneous. It is straightforward to modify our algorithms for this case.

Problem Π_T:

$$\max_{\vec{a},\vec{e}} \quad \sum_{i \in V} \alpha_i a_i P(i\text{'s data reaches the root})$$

$$\begin{aligned}
\text{s.t.} \quad & (1) a_i M + t_i T \le e_i, i \in V_L, \\
& (2) \sum_{j \in C(S)} t_j R \le e_S \\
& (3) a_i M + t_i T + \sum_{j \in C(i)} t_j R \le e_i, \text{ other nodes} \\
& (4) t_i \in \{0, 1, 2, ..., \beta_i\}, a_i \in \{0, 1\}
\end{aligned}$$

We solve Π_T as follows. Let $X[i, r, a_i]$ represent the maximum expected amount of predicted information obtained by the sub-tree rooted at node i if node i expends r units of energy on reception, and the activation state of i is a_i.

For a leaf node i, $X[i, r, a_i] = a_i \alpha_i$. For any non-leaf node, $X[i, r, a_i] =$

$$\alpha_i a_i + \max_{\sum_{j \in C(i)} t_j \le r} \sum_{j \in C(i)} X[j, \lfloor \frac{e_j - t_j T - a_j M}{R} \rfloor, a_j] f_j(t_j) \tag{2}$$

If node i is used for sensing, then only an energy of $rR - M$ is available at node i for receptions, and node i provides an information of α_i. If node i is not used for sensing, an energy of rR is available for receptions but node i does not provide any information.

For a given node i, and for a given r, we now have to compute two values of X, where $X[i, r, 1]$ corresponds to the case where i is selected as a source, and $X[i, r, 0]$ corresponds to the case where i is not selected as a source. Further, in order to compute $X[i, r, a_i]$, we also need to determine which of the children of i are selected as sources. While a brute force approach for k children requires selecting from 2^k choices, we can extend Algorithm 1 to obtain a low complexity solution to this problem. The following algorithm shows how to compute $X[i, r, a_i]$ for a given i, r, and a_i. Let $C(i) = \{C_1, ..., C_k\}$.

Table 3: Algorithm 2

1	Assign an interval $[a_j, b_j]$ to each child C_j such that $a_1 < b_1 < a_2 < b_2 < ... < a_k < b_k$.
2	Modify graph G' (Lemma 5.3) as follows. Since each child has two possibilities (to be a source or not), construct two nodes for each node in G', one representing the case where the child is a source, and the other where it is not a source. The two nodes will have different weights, one corresponding to $X[\cdot, \cdot, 1]$ and the other corresponding to $X[\cdot, \cdot, 0]$.
3	Construct a rectangle graph by representing each node as a cross-product of the interval corresponding to the child it represents and the interval corresponding to its energy allocation.
4	Find an MWIIS in this graph.

Theorem 6.1. *Algorithm 2 solves $X[i, r, a_i]$ for any node i, and for any permissible value of r and a_i.*

Proof. The proof is similar to that of Theorem 5.5. The key difference between the algorithms is that we now have two rectangles (with two different weights) in this problem for each rectangle in Theorem 5.5. One of the rectangles corresponds to the case where the child is a source and the other corresponds to the case where the child is not a source. We can only select one of these rectangles at most since a child cannot both be a source and not a source at the same time. Clearly, since both these rectangles have the same coordinates, only one of them can be chosen in the

IIS. Therefore, by finding an MWIIS, we obtain the optimal solution to Equation (2). Thus, Algorithm 2 is optimal. □

We can now use Algorithm 1 in conjunction with Algorithm 2 to determine an optimal solution to problem Π_T. This provides the optimal set of sensors for sensing the target during the next sampling instant, and the optimal transmission and reception energy to be spent by each sensor in the network.

7. NUMERICAL RESULTS

In this section, we investigate two interesting heuristic algorithms, and study how our optimal solution performs compared to these algorithms. For the results below, we use retransmissions as the error recovery scheme. Specifically, if a packet gets lost, a node can retransmit up to a certain number of times in order to improve the reliability of the link. However, as mentioned earlier, our problem setting and solutions are quite general and applicable to other types of transmission strategies that allow for coding as well. We consider the tree in Figure 6(a) with the black node as the root. The probability of error of each link is shown in the figure. All the nodes in the tree except the root are source nodes. We initially assume that the information provided by a source node is one. We also assume that each transmission/reception takes one unit of energy. We consider Definition 1 here. For a link (i, j), we take $f_i(t_i) = (1 - p_{ij}^{t_i})$ where p_{ij} is the probability of error of (i, j). This metric provides us with the expected number of source nodes whose data reaches the root. This is because for independent errors, the probability that a packet is successful over link (i, j) if t_i retransmissions are allowed is $(1 - p_{ij}^{t_i})$. Since a source node accounts for unit information if successful, and none otherwise, Definition 1 provides the expected number of source nodes whose data reaches the root.

Consider the heuristic algorithms (H_1 and H_2) below.

No retransmissions (H_1): Most existing works [3], [12] assume that links are error-free. Therefore, they do not account for retransmissions. In this case, each node can make at most one transmission.

Equal energy (H_2): This heuristic allows for retransmissions but does not take the link errors or node weights into account. Instead it equally splits the available energy at each node starting from the root to the leaves. For instance, assume that the per-node energy limit is 10 units for all nodes in the network. Starting from the root, the root's children can make up to 5 transmission each. Since 5 units of energy is allocated for transmissions, the root's children can only allocate 5 units of energy for reception from their children. This is split equally among the children. This process continues down the tree till the leaves. This heuristic is very simple to implement. However, if the per-node constraints are not identical across nodes, many nodes may not be able to make a transmission.

7.1 Maximum information

Figure 6(b) shows how H_1 and H_2 compare with our optimal solution. We assume that the per-node energy constraint is the same for all the nodes in the tree. Clearly, the optimal solution performs better than the heuristics. Consider H_1. We see that when the per-node energy constraint is low, the performance of this heuristic is close to that of the optimal solution. However, when the energy constraint is higher, this heuristic cannot make use of the additional energy available, and hence the information is quite poor for high energy constraints. On the other hand, H_2 performs close to the optimal solution when the per-node energy constraint is higher. The reason is that $f_i(t_i) = (1 - p_{ij}^{t_i})$ is a concave function of t_i. Therefore, we get diminishing returns as t_i increases. So this solution is close to the optimal solution for high energy constraints. However, when the energy constraint is low, we can see that H_2 performs poorly compared to both the optimal solution, and H_1.

7.2 Non-uniform weights and link errors

In the above experiments, the information provided by each source node is one. Further, while the links are unreliable, the probability of errors of links are close to each other. This is the reason why the heuristics perform close to the optimal solution for particular values of the energy constraint. We now modify the information provided by the red node in the tree to 100, and the probability of error of the link from the red node to its parent from 0.15 to 0.6. Figure 6(c) now shows the maximum information obtained for various energy constraints. It can be clearly seen that both H_1 and H_2 perform poorly for all values of the energy constraints when compared to our optimal solution. As before, H_1 still performs better than H_2 when the energy constraints are low, and the reverse happens when energy constraints are high. However, compared to Figure 6(b), we can clearly see that our optimal solution is significantly better than the heuristics for all values of the energy constraint.

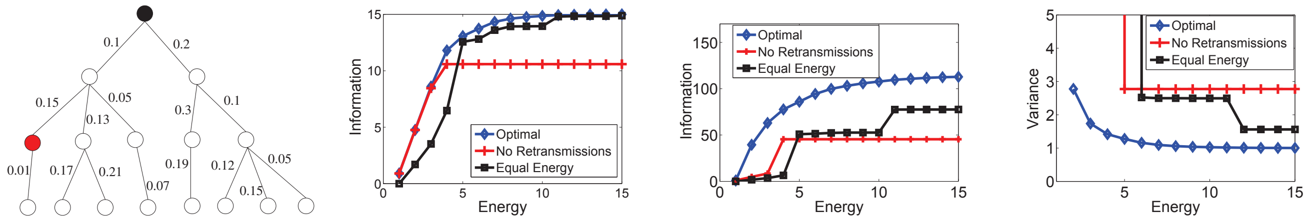
7.3 Sensor selection

We finally evaluate our sensor selection algorithm. We use the same tree as before. We now assume that each sensor has a random measurement error variance between 0 and 10, and that the inverse of the overall error variance is the sum of the inverses of the individual error variances. For the heuristics, we first allocate available energy for sensing (which costs one unit), and the remaining energy for communication. Figure 6(d) compares the overall error variance that our algorithm obtains with that of the heuristics. We can observe that the overall variance decreases as the energy constraint increases. When the energy constraint is low (< 5), our algorithm outperforms both the heuristics. Even when the constraint is high, the optimal solution performs significantly better than the heuristics.

Thus, we observe the advantages of explicitly considering unreliability in our optimization framework. Since the optimal solution is distributed, and has a low computational complexity, it is worthwhile in using this solution to tackle unreliability in links.

8. CONCLUSION

In this paper, we have investigated the problem of aggregated convergecasting in unreliable wireless networks with per-node energy constraints. We developed an optimization framework based on integer programming that explicitly accounts for unreliable links and per-node energy constraints. This framework also allows for using a general error-recovery mechanism as long as errors across links are independent. We have developed a low complexity, distributed optimal solution based on finding an MWIIS in rectangle graphs. We also studied an application of this algorithm to track targets in wireless sensor networks. Finally, we provided numerical



(a)Tree for Simulation (b)Information (c)Non-Uniform Weights (d)Sensor Selection

Figure 6: Numerical Evaluations

results that showed the importance of including unreliability in the problem framework. Future work involves accounting for more general fusion functions, and a general energy metric for fusing information from multiple nodes as this could potentially increase the size of the packet that is being transmitted, and may hence require more energy.

9. ACKNOWLEDGEMENTS

This work was supported in part by ARO MURI Awards W911NF-07-10376 (SA08-03) and W911NF-08-1-0238, and NSF Awards 0626703-CNS, 0635202-CCF, and 0721236-CNS.

10. REFERENCES

[1] R. Bar-Yehuda and M. M. Halldórsson. Scheduling split intervals. *SIAM J. Comput.*, 2006.
[2] J. Bermond, L. Gargano, and A. A. Rescigno. Gathering with minimum delay in tree sensor networks. *LNCS*, 2008.
[3] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola. On the use of binary programming for sensor scheduling. *IEEE Transactions on Signal Processing*, 2007.
[4] C. Florens, M. Franceschetti, and R. McEliece. Lower bounds on data collection time in sensory networks. *IEEE JSAC*, 22(6), 2004.
[5] A. Giridhar and P. R. Kumar. Computing and communicating functions over sensor networks. *IEEE JSAC*, 2005.
[6] A. Goel and D. Estrin. Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk. In *SODA*, 2003.
[7] S. Hariharan and N. B. Shroff. Maximizing aggregated revenue in sensor networks under deadline constraints. In *IEEE CDC*, 2009.
[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient Communication Protocols for Wireless Microsensor Networks. In *International Conference on System Sciences*, 2000.
[9] V. Isler, J. Spletzer, S. Khanna, and C. J. Taylor. Target tracking with distributed sensors: the focus of attention problem. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.
[10] C. Joo, J. Choi, and N. B. Shroff. Delay performance of scheduling with data aggregation in wireless sensor networks. In *IEEE INFOCOM*, 2010.
[11] D. Joseph, J. Meidanis, and P. Tiwari. Determining dna sequence similarity using maximum independent set algorithms for interval graphs. *Lecture Notes in Computer Science*, 1992.
[12] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 2009.
[13] B. Krishnamachari, D. Estrin, and S. B. Wicker. The impact of data aggregation in wireless sensor networks. In *ICDCSW '02*, 2002.
[14] H. Lee and A. Keshavarzian. Towards energy-optimal and reliable data collection via collision-free scheduling in wireless sensor networks. In *IEEE INFOCOM*, 2008.
[15] S. H. Low and D. E. Lapsley. Optimization flow control, I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 1999.
[16] H. Rowaihy, M. P. Johnson, O. Liu, A. Bar-Noy, T. Brown, and T. La Porta. Sensor mission assignment in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2010.
[17] H. L. V. Trees. *Detection, Estimation, and Modulation Theory: Part I.* John Wiley & Sons, 1968.
[18] J. L. Williams, I. J. W. Fisher, and A. S. Willsky. Approximate dynamic programming for communication-constrained sensor network management. *IEEE Transactions on Signal Processing*, 2007.
[19] Y. Wu, S. Fahmy, and N. B. Shroff. On the construction of a maximum-lifetime data gathering tree in sensor networks: Np-completeness and approximation algorithm. In *IEEE INFOCOM*, 2008.
[20] Z. Ye, A. Abouzeid, and J. Ai. Optimal policies for distributed data aggregation in wireless sensor networks. In *IEEE INFOCOM*, 2007.
[21] Y. Yu, B. Krishnamachari, and V. K. Prasanna. Energy-latency tradeoffs for data gathering in wireless sensor networks. In *IEEE INFOCOM*, 2004.
[22] H. Zhang, A. Arora, Y. Choi, and M. G. Gouda. Reliable bursty convergecast in wireless sensor networks. In *ACM MOBIHOC*, 2005.