Energy-aware Scheduling with Deadline and Reliability Constraints in Wireless Networks

G. Sudha Anil Kumar, G. Manimaran and Z. Wang
Real-Time Computing and Networking Laboratory
Dept. of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011
Email: {anil,gmani,zhengdao}@iastate.edu

***Abstract*—In this paper, we address the problem of scheduling a set of periodic real-time messages in a wireless network with the objective of minimizing the total energy consumption while meeting deadline and reliability constraints. We formally prove that this problem is NP-hard and solve it in two stages. First, we consider a simple model that assumes that the wireless channel is completely reliable and the network is fully provisioned. Using the technique of employing multiple hop-by-hop transmissions instead of a single direct hop transmission as the basis, we prove that the strategy of choosing the hop distances such that they are equidistant is optimal in terms of energy consumption under the deadline constraint. Based on the intuition provided by the optimal strategy, we present heuristic scheduling algorithms for a more realistic wireless channel model and network condition. Our simulation results show that the proposed scheduling algorithms provide significant energy savings over the baseline algorithms.**

I. INTRODUCTION

Wireless technology has come a long way from early days of providing minimal QoS (Quality of Service) broadcast radio services to current state of the art multimedia services on embedded mobile devices. While the current generation wireless networks are seamlessly providing users with soft real-time multimedia and data services, the efforts to develop more deterministic real-time applications on wireless networks are constantly increasing. However, typical energy constraints associated with battery driven wireless devices and the time varying nature of the wireless channels have been the major hindrances. Several energy-aware protocols have been proposed to address the challenge of providing deterministic and reliable real-time services over the wireless networks. These protocols can be broadly classified into two classes based on the kind of service they provide:

- *Throughput oriented*: This class of protocols exploit the tradeoff between energy consumption and network throughput. In [1], a protocol framework has been proposed for multi-hop wireless networks to maximize the throughput while keeping the energy costs low. These protocols primarily target different non-real-time applications where timely message delivery is not of paramount importance.
- *Soft real-time*: This category of protocols try to minimize either the latency or deadline violations in a best effort manner. The protocols which aim at minimizing the latency [2], [4] cannot provide any kind of guarantees and are unsuitable for real-time applications. On the other hand, the existing protocols which try to minimize the deadline violations either ignore the time varying nature of the wireless channel or try to minimize the deadline violations in a best effort manner. In [5], a centralized real-time MAC (Medium Access Control) protocol is presented which works over the Hybrid Coordination Function (HCF) protocol of IEEE 802.11 with the aim of facilitating long sleep durations for the individual nodes. The authors proposed a non-preemptive EDF (Earliest Deadline First) based algorithm for message scheduling in the MAC layer assuming a non-time-varying channel. In [6], an energy aware data gathering protocol is presented for wireless sensor networks which uses dynamic modulation scaling technique to minimize the energy consumption while meeting the deadlines. Both the above protocols do not provide any per message guarantees and when applied to a typical time varying channel, the proposed algorithms might lead to unpredictable number of deadline violations.
  Some protocols provide more precise guarantees like minimum deadline success ratio which is defined as the fraction of the input messages that are guaranteed to meet the deadline. In [7], the problem of scheduling multiple video streams has been considered where each stream is guaranteed a minimum deadline success ratio while minimizing the total energy consumption. These protocols are suitable for multimedia applications where occasional deadline violations are tolerable; however, they cannot provide per message guarantees and hence cannot be directly applied to real-time applications.

Providing hard deadline guarantees in a wireless environment is practically infeasible due to the time varying and lossy nature of the wireless channel[10]. Therefore, allocating a certain amount of resources (e.g. time slots) to a deadline constrained message ignoring the channel conditions may not always result in a successful and timely message delivery. Moreover, the probability of timely message delivery increases with increasing the resources allocated to the message. Therefore, resource allocation should be done both based on the required probability of successful message delivery as well as considering the nature of the channel.

In this paper, we consider hard real-time applications like industrial automation or real-time wireless sensor networks where a per message guarantee is critical. For every message, we associate a **reliability constraint** with the conventional deadline (latency) constraint. Specifically, if a message, m_i has a reliability constraint of R and a deadline of T_i, this means that every instance of m_i should reach the destination before the deadline T_i with a minimum probability of R. We stress that this kind of a deadline guarantee is much stronger than a pure best effort approach as it provides a probabilistic measure for successful and timely message delivery.

Energy consumption becomes a critical factor when trying to provide such hard guarantees on a per message basis. Over allocating resources to attain high reliabilities can result in intolerably high energy consumptions which can subsequently bring the network down. *Therefore, the three parameters: deadline, reliability and energy consumption should be considered together in order to provide hard guarantees over a wireless network, which is the focus of the paper*. The protocols presented in [3], [4] consider the three parameters together however, they provide latency and reliability guarantees in a best effort manner and hence are unsuitable for hard real-time applications.

For a given reliability constraint, it is possible to trade-off time for energy through hop-by-hop transmissions. In a wireless network several source-destination paths may be satisfying the reliability constraint while some of them are high energy consuming paths incurring less time and the rest are low energy paths taking more time. In this paper, we use the lesser energy consuming hop-by-hop transmission as opposed to a single direct transmission while providing hard per message guarantees. Most of the existing work [11], [12] which uses hop-by-hop transmissions primarily addressed energy aware throughput optimization problems which cannot be directly used for the scheduling of messages with deadline and reliability constraints.

Our **basic approach** is as follows: Instead of performing message transmissions via the long distance hop connecting the source and destination, it is possible to reduce the energy consumption if the message is transmitted via multiple hops with each hop being smaller than the direct hop. Such an approach would reduce the energy consumption while incurring more time as each hop communication would incur an entire time slot in the shared wireless medium. In the rest of the paper, we analyze this tradeoff between time and energy. The following motivational example illustrates the time-energy tradeoffs and provides further insights into the problem.

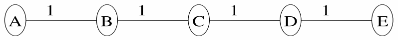


Fig. 1. A simple linear network

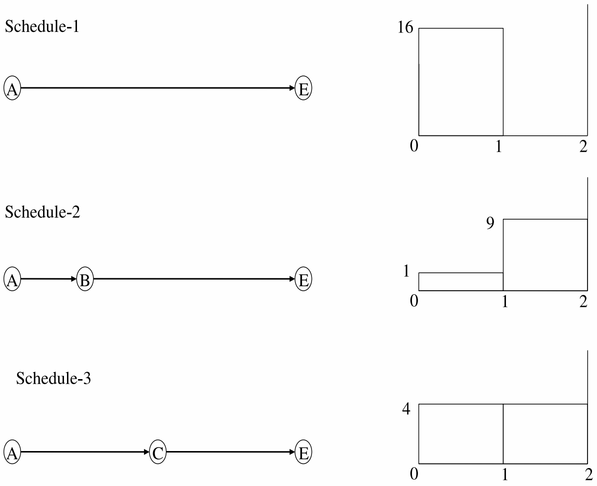


Fig. 2. Different hop-by-hop schedules

Consider a simple linear network shown in figure 1 where each hop is of unit distance, we assume each transmission takes a unit time. Further consider a single message $m_1(A,E)$ that needs to be scheduled which has a deadline of 2 units. The message can take a maximum of two hops for reaching the destination as more than two transmissions can not be accommodated before the deadline. One single hop path and several two hop paths exist between the source node A and destination node E. Figure 2 shows three schedules each using a different path with different intermediate nodes and hence different hop lengths. In the schedules shown, the y-axis denotes the transmission power defined as the square of the hop length and the x-axis denotes the time. The energy consumed in transmitting over a hop of length h is proportional to $h^2\alpha$, where α is the time taken for each hop transmission. For simplicity, we assume that $\alpha = 1$ in this example. The energy consumptions of the three schedules are 16 $(= 4^2*1)$,10 $(= 1^2 * 1 + 3^2 * 1)$ and 8 $(= 2^2 * 1 + 2^2 * 1)$ respectively. This example illustrates the fundamental tradeoff between time and energy offered by the hop-by-hop transmissions. Larger hops consume fewer time slots while incurring higher energies. For example, schedule 1 chooses the longest hop consuming just one time slot but incurs highest energy among the three schedules. Schedules 2 and 3 choose relatively smaller hops incurring lesser energy while consuming two time slots instead of one. Therefore, choosing smaller hops saves energy. However choosing too a small hop will force larger hops to be taken at a later point in the schedule in order to reach the destination before the deadline. Such an unequal choice of hops in non-ideal from energy savings point of view. For example, comparing schedules 2 and 3, schedule 2 uses two unequal hops ($AB < BE$) while schedule 3 uses hops of equal length($AC = CE$). The convex nature of the transmission power function favors such **equal hopping**. In fact, this approach of choosing equal hop lengths is optimal whenever the network topology allows such equal hopping. We prove this in the subsequent sections.

The rest of the paper is organized as follows: section II presents the problem statement followed by our system model. Section III presents the optimal solutions for a simple model. Section IV presents energy-aware scheduling algorithms for the general model. Section V and VI present simulation results and conclusions respectively.

II. PROBLEM FORMULATION

Energy Aware Real-Time Message Scheduling (EARTMS) Problem: Consider a set of periodic messages $(m_1, m_2, ..., m_M)$ where each message m_i has a period T_i equal to its deadline. The symbols src_i and $dest_i$ denote the source and destination nodes of message m_i in the network. Let $path_{i1}, path_{i2} ... path_{ij} ... path_{iN_i}$ denote the different possible paths between src_i and $dest_i$. The objective of the EARTMS problem is to choose one of the N_i paths for each message m_i which minimizes the total energy consumption and satisfies both reliability and deadline constraints. When dealing with periodic messages, the total energy consumption is defined with respect to a time window, say $[0, T_o]$, where T_o is an application specific parameter. For a given T_o, the problem can be mathematically stated as follows:

Minimize
$$\sum_{i=1}^{M}\sum_{j=0}^{N_i}\lceil\frac{T_o}{T_i}\rceil E_{ij}x_{ij} \tag{1}$$

Subject to:
$$\sum_{i=1}^{n}\frac{\sum_{j=0}^{N_i}\delta_{ij}x_{ij}}{T_i} \leq 1 \tag{2}$$

$$\sum_{j=0}^{N_i} Rel_{ij}x_{ij} \geq R, \forall i \in [0, M] \tag{3}$$

$$\sum_{j=0}^{N_i} x_{ij} = 1, \forall i \in [0, M] \tag{4}$$

$$x_{ij} \in \{0, 1\}, \forall i, j \tag{5}$$

Where, E_{ij} is defined as the total energy consumption incurred by the transmission of m_i over $path_{ij}$; δ_{ij} and Rel_{ij} denote latency incurred and reliability offered by $path_{ij}$ respectively. The binary variable x_{ij} indicates the selected path for m_i. The objective function in equation (1) represents the total energy consumed by the message set in a time window of $[0, T_o]$. The constraint in equation (2) ensures that the message set remains schedulable under a work conserving scheduler like EDF. Equation (3) represents the per message reliability constraint. The binary variable x_{ij} indicates the selected path for m_i and the constraint in equation (4) ensures only one of the N_i paths is chosen for each message m_i.

The above EARTMS problem is hard to solve and has a large solution space. We make the following two simplifying and practically achievable transformations on the EARTMS problem to ease the analysis. We further prove that the transformed version of the EARTMS problem is NP-hard.

- Period Transformation: The EARTMS problem addresses the periodic message workload with each message having a distinct period. However, analysis becomes much easier when the periods of all the messages are equal. Therefore, we use the idea of period transformation to make the message periods equal with appropriate scaling in transmission times. For example, consider two messages with periods of 4 and 8, respectively; let their respective transmission times be 2 and 4. Applying period transformation will result in three messages (message 2 is split into two) with each having a period of 4 and transmission time of 2. In the rest of the paper, we assume the period transformation is made apriori and all messages have a common deadline of T equal to T_o.
- Equal Link Reliability Assignment: We assume that each link in the network is designed to offer a fixed reliability denoted by R_L. For a given link, depending on its channel parameters the appropriate number of retransmissions required to offer a reliability of R_L can be calculated which will govern the link latency. We show these calculations in section 4. The R_L is a design parameter which is generally chosen to be high. With this transformation, the reliability constraint is now transformed to a hop-constraint denoted by k. In other words, every path which has less than or equal to k hops is a reliable path where, k is the largest integer satisfying the relation $R_L^k \geq R$.

With these transformations in place, in the rest of the paper, we solve the transformed EARTMS problem which is mathematically restated as follows:

Minimize
$$\sum_{i=1}^{M}\sum_{j=0}^{N_i'} E_{ij}x_{ij} \tag{6}$$

Subject to:
$$\sum_{i=1}^{M}\sum_{j=0}^{N_i'} \delta_{ij}x_{ij} \leq T \tag{7}$$

$$\sum_{j=0}^{N_i'} x_{ij} = 1, \forall i, j \tag{8}$$

$$x_{ij} \in \{0, 1\} \tag{9}$$

Where N_i' denotes the number of paths for message m_i each having fewer than k hops.

The transformed EARTMS problem is NP-hard and we prove this by reducing the well known Multiple Choice Knapsack problem to the problem at hand. The MCKP problem can be stated as follows.

Multiple Choice Knapsack Problem (MCKP): Given λ classes $N_1, ... N_\lambda$ of items to pack in a knapsack of capacity c. Each item $O_{ij} \in N_i$ has a profit p_{ij} and weight w_{ij}, and the problem is to choose one item from each class such that the profit sum is maximized without having the weight sum to exceed c [13]. Mathematically,

Maximize

$$\sum_{i=1}^{\lambda} \sum_{j \in N_i} p_{ij} x_{ij} \tag{10}$$

Subject to:

$$\sum_{i=1}^{\lambda} \sum_{j \in N_i} w_{ij} x_{ij} \leq c \tag{11}$$

$$\sum_{j \in N_i} x_{ij} = 1 \tag{12}$$

$$x_{ij} \in \{0, 1\} \tag{13}$$

Theorem 1: The transformed EARTMS problem is NP-hard.
Proof: For the convenience of reduction, we rewrite the maximization objective of the MCKP as the following minimization objective.

$$\sum_{i=1}^{\lambda} \sum_{j \in N_i} (1 - p_{ij}) x_{ij} \tag{14}$$

Now, the MCKP has a one to one correspondence with the EARTMS problem and can be reduced to an instance of it as follows. Create a message m_i for each of the classes N_i and for each object O_{ij} in N_i create a path $path_{ij}$ with exactly k hops between src_i and $dest_i$. Further, assign an energy consumption of E_{ij} equal to $(1-p_{ij})$ and a latency δ_{ij} equal to w_{ij} to $path_{ij}$. This reduction takes $O(\sum_{i=1}^{\lambda}(\mid N_i \mid)k)$ amount of time which is clearly polynomial of the problem size. This implies that if the EARTMS problem can be solved in polynomial time, then the MCKP can also be solved in polynomial time. However, it is known that the MCKP problem is NP-hard [13]. Therefore, the EARTMS problem is also NP-hard. Further, similar to the MCKP, given a solution to the EARTMS problem it is not easy to verify it in polynomial time. Therefore the EARTMS problem is indeed NP-hard. □

In the following we present our system model. Table I lists all the symbols that are used in the rest of the paper.

Symbol	Description
n	Total number of nodes in the network
M	Total number of input messages
T	deadline expressed as number of slots
α	duration of each time slot
β	overhead (time slots) of the DHLA algorithm
σ	ratio of dynamic slack to β
	Channel model symbols
P_{gg}	Good state probability
P_{bb}	Bad state probability
P_{sgood}	Steady state probability of good state
P_{sbad}	Steady state probability of bad state
	Message specific symbols
m_i	Message from node src_i to node $dest_i$
src_i	source node of m_i
$dest_i$	destination node of m_i
$path_{ij}$	jth path of message m_i from src_i to $dest_i$
N_i	number of feasible paths from src_i and $dest_i$
R	Required per message reliability
a_i	number of slots allocated to m_i
s_i	number of slots actually used my m_i
	Path specific symbols
δ_{ij}	Latency incurred by m_i when transmitted over path $path_{ij}$
Rel_{ij}	Reliability offered to m_i when transmitted over path $path_{ij}$
E_{ij}	Energy incurred by a transmission of m_i over path $pathij$
k	maximum number of hops allowed in any path
	Link/hop specific symbols
δ_i	Latency of the direct link connecting src_i and $dest_i$
$P_{min}(i,j)$	Minimum required transmission power over the link (i,j)
$D(src, dest)$	physical distance between nodes src and $dest$
D_i	equal to $D(src_i, dest_i)$
R_L	reliability that each link is designed to offer
r	maximum number of retransmissions allowed is (r-1)
h	physical distance of a link/hop in the context

TABLE I
NOTATION

A. System Model

1) Network Model: We consider a wireless network with n nodes which share the common wireless medium that is accessed in an exclusive manner. A source node can communicate with its destination node either directly or by making several hop-by-hop transmissions through intermediate nodes. We assume that the shared wireless medium is slotted in time with a slot size of α and all nodes are time synchronized. One of the nodes acts as a centralized scheduler which constructs the schedule and broadcasts it in the network. All messages are of equal length and take α units of time for a transmission irrespective of the source and destination.

2) Channel Model: Although all the nodes in the network are in the same interference range and share the wireless medium, there is a high likelihood for one link connecting a pair of nodes to be in bad state while some other link is in a good state. Therefore, we assume that each wireless link is independent of the other. We model the time varying channel behavior using the Gilbert-Elliot model [14]. A Gilbert-Elliot channel transitions between the good and bad states at the end of each time slot as per the state transition diagram shown in figure 3. It can be viewed as a two state discrete time Markov chain. The steady state probability of a particular Markov state (good or bad in this case) is defined as the probability that the Markov chain is in that state after arbitrarily large number of state transitions. The steady state probabilities for this model can be calculated as follows:

$$P_{sgood} = \frac{(1 - p_{bb})}{(2 - p_{gg} - p_{bb})} \tag{15}$$

$$P_{sbad} = \frac{(1 - p_{gg})}{(2 - p_{gg} - p_{bb})} \tag{16}$$

These probabilities are used in section 5 to estimate the link reliabilities.

3) Power Model: For every wireless link (i, j), there is a minimum transmission power that is required to transmit from i to j which is denoted by $P_{min}(i, j)$. All messages that are transmitted at a power greater than $P_{min}(i, j)$ are successfully received if the corresponding link (i, j) is in good state. Messages that are transmitted at a power lower than $P_{min}(i, j)$ can be error prone even when the channel is in good state. The $P_{min}(i, j)$ values depend on the distance $(D(i, j))$ between nodes i and j. Mathematically,

$$\frac{P_{min}(i,j)}{P_{min}(x,y)} = \frac{D^{\gamma}(i,j)}{D^{\gamma}(x,y)} \tag{17}$$

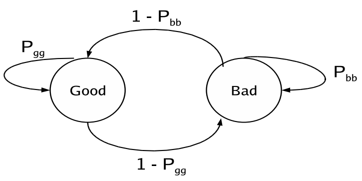


Fig. 3. Gilbert-Elliot Model

Where γ is the attenuation factor and typically belongs to the range of $[2, 6]$. In this paper, we assume $\gamma = 2$. Our results however, hold for higher values as well.

By default, all messages over a link (x, y) are transmitted at $P_{min}(x, y)$ power and the energy incurred in transmitting a single message over a link/hop of length h (equal to the distance $D(x, y)$)is calculated as αh^2. In the rest of the paper we use the terms link and hop interchangeably.

4) Scheduling Model: We follow a two stage (offline and online) preemptive scheduling model. In the offline phase, the proposed scheduling algorithms assign hop-by-hop paths to individual messages ensuring that the message set remains schedulable under EDF [15]. In the online phase, an EDF scheduler is employed which schedules the messages in the channel considering the modified message latencies due to updated paths. We assume the online scheduler follows a FIFO policy while scheduling messages with equal deadlines. Further, the input message set is assumed to be schedulable when each message is transmitted over the direct link connecting its source and destination.

III. SCHEDULING RESULTS

In this section, we derive an optimal scheduling strategy for a simple model and relax the simplifying assumptions one by one as we proceed.

A. Scheduling results for a simple model

The following two assumptions define our simple model, in which we will prove that the equi-distant hopping is an optimal strategy.

- **Continuity Assumption:** For each message m_i, this assumption allows us to take arbitrary hop lengths i.e. hop length h varies continuously from 0 to D_i. Where D_i is the short form of $D(src_i, dest_i)$. In order to realize this assumption we need to have intermediate nodes at every coordinate in the network.
- **Good Channel Assumption:** All links are in good states all the time i.e., $p_{gg} = 1, p_{bb} = 0$. This assumption allows us to ignore the erroneous transmissions. Both the assumptions are artificial and we will relax both of them later in this section.

Lemma 1: If a message transmission completes before the deadline time $T\alpha$, the energy consumption is not optimal.

Proof: Let us assume the optimal path for transmitting a message m_i has p hops and incurs a time $t\alpha$, where $t < T$. Let the energy consumption of such a transmission be E. Since this is optimal, there is no other path which can further minimize the energy consumption. We will prove the lemma by contradiction. Now, let us replace the p^{th} hop h_p in the above transmission with $(T - t + 1)$ hops each of length $\frac{h_p}{(T-t+1)}$. Due to the continuity assumption, it is always possible to achieve such hop lengths by finding out the corresponding intermediate nodes. Now the total time taken by the above hop-by-hop transmission is $(t-1)+(T-t+1) = T$ and the energy consumed in this case is $E_{new} = E - \alpha h_p^2 + (T - t + 1) * \alpha * (\frac{h_p}{T-t+1})^2$. Clearly, $E_{new} < E$, which is a contradiction. Hence the proof. ☐

Intuitively, the above lemma states that the approach of using large hops to reach the destination and completing the transmission earlier than the deadline does not minimize the energy consumption of the message transmission.

Theorem 2: If a message m_i is transmitted over T hops where each hop length $h = \frac{D_i}{T}$ and the message transmission completes just at the deadline time $T\alpha$, such an equal hopping approach minimizes the energy consumption of the transmission optimally.

Proof: The total energy consumption of the equal hopping approach with each hop equal to h is given by:

$$E_0 = T\alpha h^2 = \frac{\alpha D_i^2}{T} \tag{18}$$

Let us consider an alternative approach where the message m_i is transmitted from src_i to an arbitrary intermediate node x using n_1 hops each of length h_1 and from x, it is transmitted to the node $dest_i$ using n_2 hops each of length h_2.

From lemma 1, we know that any message transmission which completes before the deadline will not minimize the energy consumption optimally therefore, the best possible schedule using hop lengths h_1 and h_2 can be obtained when the transmission completes just at the deadline. That is,

$$(n_1 + n_2)\alpha = T\alpha \tag{19}$$

Furthermore, we have:

$$n_1 * h_1 + n_2 * h_2 = D_i \tag{20}$$

The above equations hold if and only if one of the following two inequalities hold:

$$0 < h_1 < h < h_2 < D_i \tag{21}$$

$$0 < h_2 < h < h_1 < D_i \tag{22}$$

This is because if both $h_1 < h$ and $h_2 < h$ (too small hops), then the required amount of time will be greater than the deadline $T\alpha$ which is not allowed. On the other hand, if both $h < h_1$ and $h < h_2$ (too long hops), then the transmission would complete before the deadline which results in a non-optimal energy consumption as per lemma 1. The total energy consumption of the message transmission satisfying equations 19-22 is given by:

$$E_1 = n_1\alpha h_1^2 + n_2\alpha h_2^2 = \alpha(D_i(h_1 + h_2) - Th_1h_2) \quad (23)$$

Now we prove that E_0 is less than E_1 for all h_1, h_2, T and α. Consider, $E_0 - E_1$:

$$E_0 - E_1 = \frac{\alpha(D_i - Th_1)(D_i - Th_2)}{T} \quad (24)$$

From theorem statement, $D_i = Th$, therefore,

$$E_0 - E_1 = \alpha(h - h_1)(h - h_2) \quad (25)$$

which is clearly negative from the inequalities in equations 21-22. Therefore, the energy consumption is optimal when a single hop length h is used and the transmission is completed just at the deadline time. □

Theorem 3: If two messages m_i and m_j with a common deadline $T\alpha$ are transmitted via different hop-by-hop paths each consisting of hops of equal length h, given by $h = \frac{D_i + D_j}{T}$, then the total energy consumed is optimal.

Proof: Let $D = D_i + D_j$, then $h = \frac{D}{T}$. The total energy consumption of the equal hopping approach is given by:

$$E_0 = T\alpha h^2 = \frac{\alpha D^2}{T} \quad (26)$$

Consider a different approach where each message uses a different hop length. Let the first and second message be transmitted via n_1 and n_2 hops each of length h_1 and h_2 respectively. Therefore we have:

$$n_1 = \frac{D_i}{h_1} \quad (27)$$

$$n_2 = \frac{D_j}{h_2} \quad (28)$$

$$n_1h_1 + n_2h_2 = D \quad (29)$$

$$(n_1 + n_2)\alpha = T\alpha \quad (30)$$

The total energy consumption of the message transmission satisfying equations 27-30 is given by:

$$E_1 = n_1\alpha h_1^2 + n_2\alpha h_2^2 = \alpha(D(h_1 + h_2) - Th_1h_2) \quad (31)$$

From the proof of theorem 1, it is clear that $E_0 < E_1$. Hence, using a single hop length across different message transmissions would optimally minimize the total energy consumption. □

Although the results presented in this section cannot be directly applied to practical networks, they give us the insight that performing hop-by-hop transmissions with equal hop lengths would minimize the total energy consumption.

B. Relaxing the Good Channel Assumption

Our approach of handling the erroneous channels is to estimate the number of retransmissions needed for maintaining a link reliability of R_L where the **link reliability** is defined as the probability that a message is transmitted successfully over the link in at most r transmissions.

Consider the link $(src_i, dest_i)$ with the Gilbert-Elliot parameters P_{gg} and P_{bb}. The probability that the message m_i is successfully transmitted across the link in at most r transmissions (or $r-1$ retransmissions) is given by:

$$P(\delta_i \le r\alpha) = 1 - P_{sbad} * P_{bb}^{r-1} \quad (32)$$

where P_{sbad} is the steady state probability that the channel is in bad state and δ_i is the total link latency.

Given the above analysis, the latency incurred by the simple retransmission scheme in providing a required link reliability of R_L can be determined by finding out the smallest r that satisfies: $P(\delta_i \le r\alpha) \ge R_L$. For such an r, the worst-case latency incurred is given by:

$$\delta_i = r\alpha \quad (33)$$

In this paper, we assume the network is homogeneous i.e. each link has equal P_{gg} and P_{bb} values. As a consequence, all links incur the same number of retransmissions (r) for a required link reliability. Therefore, all the above theorems can proved for the realistic channel case by simply replacing α by $r\alpha$ in the analysis.

C. Relaxing the Continuity Assumption

In practice, the continuity assumption does not hold and as a result arbitrary hop lengths cannot be achieved. Nevertheless, from the analysis of the simple model, it is clear that equal hopping is a good strategy. It can be proved that an optimal solution for practical networks will choose hops as equi-distant as possible. We do not present the formal proof due to space constraints. In the subsequent sections, we present heuristic scheduling algorithms which choose paths with hops as equal as possible.

IV. ENERGY-AWARE SCHEDULING ALGORITHMS

In this section, we present heuristic solutions to the EARTMS problem starting with a simple and straight forward hop-by-hop scheduling algorithm which aggressively allocates the available slack (extra time slots) to the next message. This algorithm serves as our baseline algorithm.

A. Simple Iterative Scheduling Algorithm (SISA)

Consider an example with three messages: $m_1(A,B), m_2(C,D), m_3(E,F)$ that need to be scheduled before a deadline of $T = 6$ slots in the network shown in the top of figure 4.(a). Let us assume that each link is 100% reliabile for simplicity. A straight forward approach would schedule each message m_i via the corresponding direct hop $(src_i, dest_i)$ using a single time slot per message. We refer to this approach as the *Direct Hop Scheduling Algorithm(DHSA)* in the rest of the paper. The schedule obtained by the DHSA is shown in figure 4.(a) and the corresponding paths are shown above the schedule. The DHSA incurs an energy consumption of 48 $(= 1 * 4^2 + 1 * 4^2 + 1 * 4^2)$ leaving 3 unused slots (i.e. a slack of 3 slots). The SISA utilizes such a slack to perform hop-by-hop transmissions with each hop being smaller than the direct hop.

SISA works on an input schedule; by default this input schedule is the one produced by the DHSA algorithm. It iterates over the messages in the schedule allocating the available slack to the next message. Algorithm 1 presents the psuedocode. In step 3, the least energy consuming path for m_i with at most k hops and using s_i number of slots is obtained. Where $s_i < (a_i + s)$. In steps 4 and 5, the schedule and the remaining slack are updated. In step 6, the algorithm returns if there is no more slack to allocate.

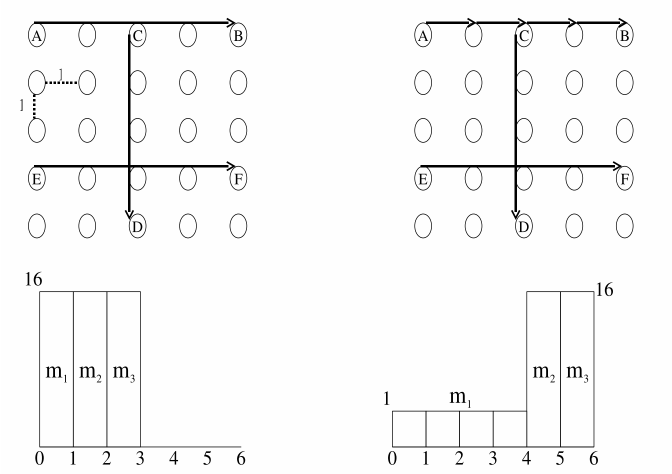
```
    Input: input schedule ($H$), slack ($s$), number of slots
           allocated to $m_i$ in schedule $H$ ($a_i$)
    Output: Output Schedule, $H_1$
1   Set $H_1 = H$;
2   foreach $i = 0$ to $M$ do
3       Find the least energy consuming path with at
        most $k$ hops for message $m_i$ in $H_1$ using a
        maximum of $a_i + s$ slots. Let the actual number
        of slots used be $s_i$;
4       Update the schedule $H_1$;
5       Update the available slack, $s = (a_i + s) - s_i$;
6       If($s \leq 0$) break;
7   end
```

Algorithm 1: Simple Iterative Scheduling Algorithm

An application of the SISA to the example at hand would result in a schedule shown in the figure 4.(b) wherein the message $m_1(A, B)$ utilizes the slack by using much smaller hop lengths of size one instead of using the direct hop of length four. The corresponding paths are shown above the schedule. This approach incurs an energy consumption of 36 $(= 4*1^2 + 2*4^2)$ resulting in 25% of energy reduction.

B. Static Hop Length based Algorithm (SHLA)

The basic idea of the ideal hop length based algorithms is to allocate slack to individual messages such that each message transmission can maintain hop lengths as close as possible to the ideal hop length in its path, where the ideal hop length is defined as the ratio of the total distance that need to be covered by all the messages to the common deadline. In this section, we present the static and dynamic ideal hop-length based algorithms. The static algorithm (SHLA) calculates the ideal hop length and constructs the schedule offline. The dynamic algorithm (DHLA) recalculates the ideal hop length on the fly utilizing the dynamic slack generated whenever a message requires fewer transmissions (less than the allocated worst-case r) over a link due to good channel condition.



(a) DHSA schedule (b) SISA schedule

Fig. 4. Schedules of the DHSA and SISA algorithms

The psuedocode for SHLA is presented in Algorithm 2. The variable D stores the total distance that needs to be covered by the remaining messages, it is initialized in step 1 and updated in step 8 after scheduling each message. Similarly, t stores the time taken by the messages scheduled so far, it is initialized in step 1 and updated in step 7. In step 3, the ideal hop length is calculated. In step 4, the maximum number of hops a message m_i can have in its path are calculated. In the following step, the SHLA ensures that the chosen path does not have too many small hops by removing all the edges smaller than h in the network graph. In step 6, the least energy consuming path between src_i and $dest_i$ with at most a_i hops is obtained and the schedule is updated in step 10. Step 12 checks if there is more slack available in which case, the algorithm calls SISA to utilize the leftover slack.

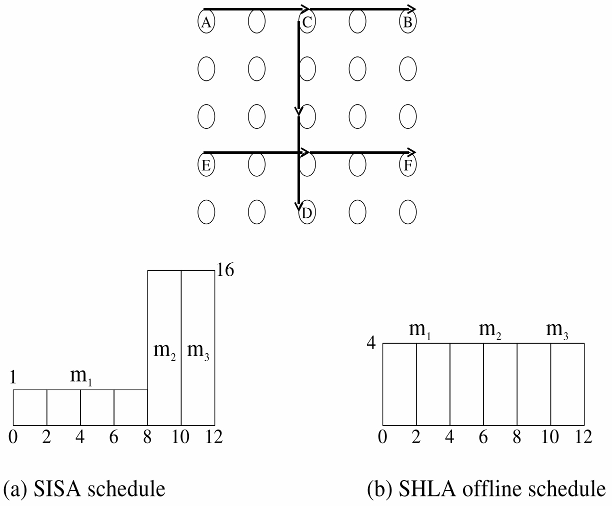
Consider an example with three messages: $m_1(A, B), m_2(C, D), m_3(E, F)$ and a common deadline of $T = 12$ slots. The network is shown in the top of figure 5, below which the SISA and the SHLA schedules are shown. Let us assume the maximum number of retransmissions allowed per link is $r = 2$ and the maximum number of hops per path is 4 due to reliability constraint. In step 3 of SHLA, the ideal hop length is calculated for the given set of messages as 2 $(= \frac{D(A,B)+D(C,D)+D(E,F)}{\frac{T}{r}})$. The smallest 2-hop paths for each message are obtained as shown in the same figure. Since there is no slack left out the algorithm terminates here. The SHLA schedule results in about 33% energy reduction by incurring $48 (= 2^2 * 12)$ units as compared to $72 (= 1^2 * 8 + 4^2 * 4)$ units incurred by the SISA schedule.

C. Dynamic Hop Length based Algorithm (DHLA)

Although, SHLA reduces the energy consumption by maintaining hop-lengths closer to the ideal hop length it implicitly assumes that each hop would require maximum number of retransmissions which is a conservative assumption. There is some slack generated during the schedule when some of the links require lesser retransmissions than r. For example,

```
Input: number of messages ($M$), deadline ($T\alpha$)
Output: Output Schedule, $H_2$
set $D = \sum_{i=0}^{M} D_i$ and $t = 0$ ;
foreach message $m_i$ do
    Calculate the ideal hop length $h = \frac{D}{\frac{T-t}{r}}$ ;
    Set $a_i = min(k, \frac{D_i}{h})$ slots ;
    Remove all edges smaller than $h$ in the network graph ;
    Find the smallest path with link energy consumption as the edge weights using a maximum of $a_i$ hops. This path may actually use less than $a_i$ hops. Let $s_i$ be the actual number of hops in the obtained path. ;
    $t = t + s_i r$ ;
    set $D = D - D_i$ ;
    Add the removed edges in the network graph ;
    Update the SHLA schedule, $H_2$ ;
end
if $(T - t) \leq 0$ then
    Sort the tasks in schedule $H_2$ in the order of their energy consumption with $m_1$ being the largest energy consumer;
    SISA($H_2$,$(T - t)$);
end
```

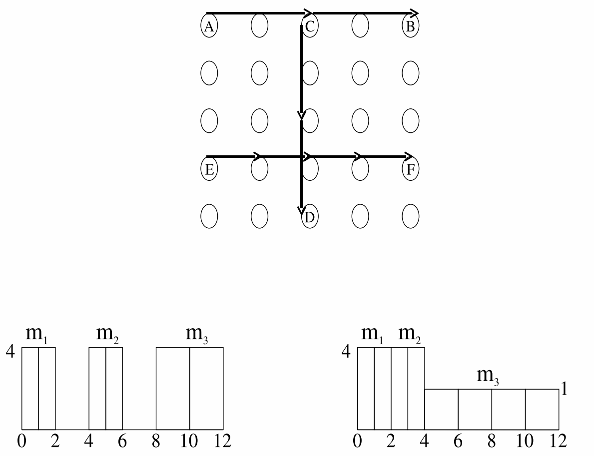
Algorithm 2: Static Hop Length based Algorithm



(a) SISA schedule

(b) SHLA offline schedule

Fig. 5. Schedules of the SISA and SHLA algorithms



(a) SHLA online schedule

(b) DHLA online schedule

Fig. 6. Online schedules of the SHSA and DHLA algorithms

consider the online schedule of the SHLA shown in figure 6.(a). The message m_1 completes in just two slots due to the good channel condition thereby leaving a dynamic slack of two slots. The DHLA utilizes such dynamic slack by updating the ideal hop-length after every message transmission. More specifically, the DHLA calculates the number of free slots generated at the end of each of message transmission i.e. after a message reaches its final destination. The centralized scheduler running the DHLA keeps track of the dynamic slack by listening to the common wireless medium and broadcasts the updated schedule corrections to all the nodes in the network. The overhead incurred by the centralized is denoted by β slots. The psuedocode for DHLA is presented in algorithm 3.

In step 1, DHLA starts with an SHLA schedule and in step 3, after each message transmission the free slots are utilized to further reduce the ideal hop length. Before calculating the new ideal hop length, the central scheduler checks in step 4 if the dynamic slack is large enough i.e. dynamic slack is at least σ times the overhead that would be incurred by the DHLA algorithm. This way, we can make sure that overhead does not overpower the energy savings offered. If the dynamic slack is not large enough the central scheduler simply broadcasts to all other nodes one to advance there schedules by $f_i - 1$ slots (that one slot is used for broadcast). Estimating the β value and deriving an optimal σ involves other system parameters (e.g, local processor speed) and needs further analysis. In this paper, we assume these values are available.

In the above example, after the first message completes there are 2 free slots which reduce the ideal hop length to $4/3$. However, there are no intermediate nodes in the network to accommodate such a hop length. Hence the schedule proceeds normally with the slots of message m_2 being preponed. We ignore the overheads in this illustration for simplicity. After the message m_2 completes its transmission, the total number of free slots is four reducing the ideal hop-length to one ($= 8/8$). Message m_3 therefore uses a hop length of one as opposed to using a hop-length of two in the original SHLA schedule. The resulting schedules are shown in figure 6. The energy consumption of the DHLA is 24 units while that of the SHLA is 32 which counts to 25% of savings.

V. SIMULATIONS STUDIES

We performed simulation studies to compare the performance of the above proposed scheduling algorithms for dif-

```
Input: number of messages ($M$), deadline ($T\alpha$)
Output: Output Schedule $H$
Schedule, $H$ = SHLA(M,T). Let the number of slots assigned to each message $m_i$ be $s_i$ in $H$;
foreach message $m_i$ do
    Listen to the completion of $m_i$'s transmission and calculate the leftover free slots as $f_i$ ;
    if $f_i > \sigma * \beta$ then
        $T = T - s_i + f_i$ ;
        $H$ = SHLA($M - i$,$T$) ;
        Broadcast the updated schedule $H$;
    end
    else if $f_i - 1 > 0$ then
        Prepone the schedule $H$ by ($f_i - 1$) slots and broadcast it;
    end
end
```

Algorithm 3: Dynamic Hop Length based Algorithm

ferent network conditions. We considered a circular region with a diameter of 1000 meters and varied the number of nodes, n which are uniformly distributed at random positions in the region. Each of the n^2 channels were independently simulated as per the P_{bb} and P_{gg} values. We assumed n^2 input messages each of size $1KB$ and the bandwidth is assumed to be $1Mbps$. Throughout the simulations, we maintained a link reliability of 0.99 and each message needed an end-to-end reliability of 0.90. We have chosen $\sigma = 2$ and $\beta = 3$ slots for all our simulations. The following parameters were varied: P_{gg}, P_{bb}, n (the number of nodes) and the slack factor, S_f defined as follows: $S_f = (deadline - Mr)/deadline$, where M is the number of messages and r maximum number of retransmissions per link. The performance metric for all our simulations was the normalized energy consumption where all energies are normalized with respect to DHSA. A hundred simulation runs were made for each set of parameters and the obtained average is plotted as a single point in the graphs.

A. Results

1) Effect of the slack factor: In this set of simulations, the slack factor is varied from 0 to 1 while choosing the other parameters as follows: $n = 40$,$P_{gg} = 0.98$, $P_{bb} = 0.80$. Figure 7 shows the relative performance of the three schemes. At $S_f = 0$, there is no scope for performing any hop-by-hop transmissions. Hence, all the three schemes would default to the direct hop transmissions thereby incurring a normalized energy consumption of 1. As the slack increases hop-by-hop transmissions become possible and the schemes perform better. This aggressive behavior of SISA results in very small hops for certain messages while certain other messages use large hops. On the other hand, the SHLA allocates slack based on ideal hop length and as a result, it performs better than the SISA through out the range. SHLA shows an average improvement of 30% over SISA. The DHLA performs even better by utilizing dynamic slack and shows an average improvement of 40% over SISA.

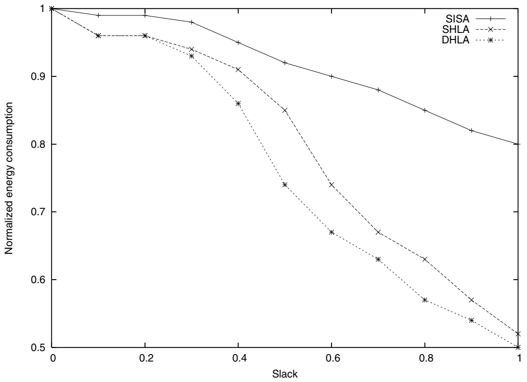


Fig. 7. Effect of slack factor

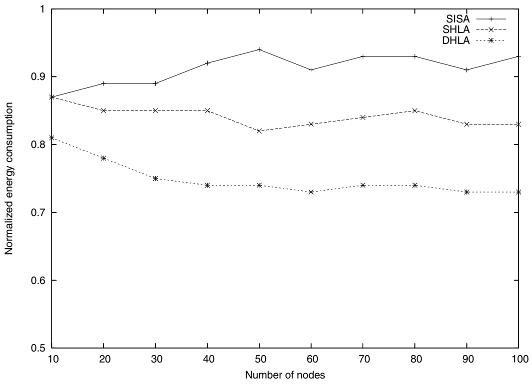


Fig. 8. Effect of number of nodes

2) Effect of number of nodes: Figure 8 shows the relative performance of the three schemes by varying the number of nodes, n. With increasing n the following two competing factors increase: the number of messages and the number of smaller hops. The energy consumption increases with increasing number of messages; on the other hand, as there are more nodes in the network each direct hop can now be broken into smaller hops with a higher chance. The effect of messages is same for all the schemes as the number of messages is constant across schemes. The relative performance of the schemes therefore depends on how effectively the hop lengths are chosen by each scheme. In the figure, at $n = 10$, there are too few nodes in the network and the average inter-node distance is large hence, the SHLA fails to find the appropriate intermediate nodes to perform ideal hop length based hop-by-hop scheduling. Consequently, it shows a behavior similar to SISA. At higher values of n, it becomes much easier to choose intermediate nodes at ideal hop lengths and hence SHLA performs better than the SISA. SHLA shows an average improvement of 15% over SISA and DHLA shows an increased average improvement of 25% over SISA.

3) Effect of good state probability: In this simulation set, we varied the P_{gg} from 0.90 to 0.99. In figure 9, with increasing P_{gg}, the number of noisy slots reduce and hence the energy consumption should reduce. This behavior is shown by both the SISA and SHLA algorithms with SHLA performing better than the SISA throughout the range. SHLA shows an average improvement of 10% over SISA. On the other hand the energy consumption of DHLA decreases until $P_{gg} = 0.92$ and increases slightly from there on. This is due to the following reason: at very low P_{gg} values, the number of retransmissions allocated to ensure an necessary amount of reliability is large which results in a lot of dynamic slack. As the P_{gg} increases lesser number of retransmissions are allocated and hence less dynamic slack thereby reducing scope for dynamic slack. DHLA shows an average improvement of 20% over SISA.

4) Effect of bad state probability: Figure 10 shows the relative performance of the three schemes with varying P_{bb}. With increasing P_{bb}, the number of actual retransmissions performed increase (the dynamic slack decreases) and hence the total energy consumption increases for all the schemes. However, the increase in energy consumption of the DHSA is typically much higher than the three schemes. Due to this reason the normalized energy consumptions appear to be decreasing at some points. SHLA performs better than SISA throughout the range showing an average improvement of 15%. Furthermore, as the P_{bb} increases, the dynamic slack decreases thereby reducing the gap between SHLA and the DHLA algorithms. DHLA shows an average improvement of 25% over SISA.

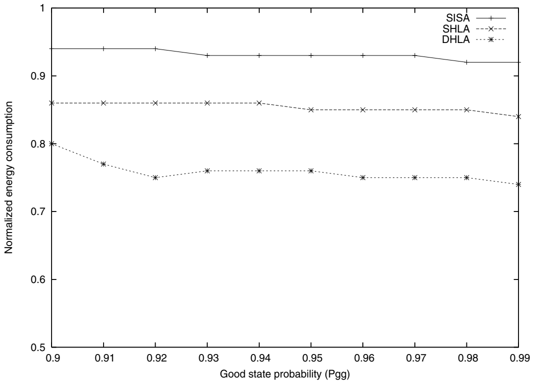


Fig. 9. Effect of P_{gg}

VI. CONCLUSION

In this paper, we considered the problem of scheduling real-time periodic messages in a wireless network with the objective of minimizing the total energy consumption while meeting the deadline and reliability constraints. We employed the technique of using less energy consuming hop-by-hop transmissions instead of high energy incurring direct hops as a basis to solve this problem. We have shown that this problem is NP-hard even after period transformation and fixed link reliability assignment. We solved this problem in two stages: in the first stage, we provided optimal solutions for a simple model and further developed effective heuristic scheduling algorithms for the general model in the later stage. Our simulation studies show that the presented algorithms provide significant energy savings over the baseline algorithms. In our future work, we plan to extend the presented scheduling algorithms to a multi-hop setup where the parallelism offered by non-interfering message transmissions can be exploited.

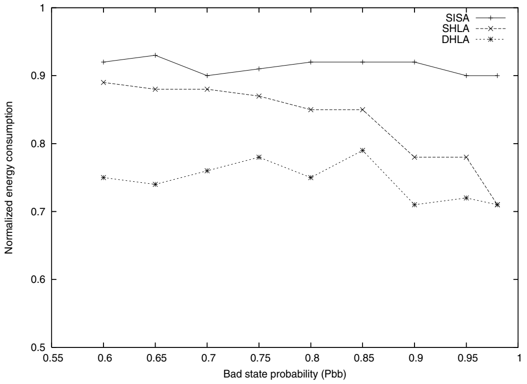


Fig. 10. Effect of P_{bb}

REFERENCES

[1] J. S. Pathasuntharam, A. Das and P. Mohapatra, "A Flow Control Framework for Improving Throughput and Energy Efficiency in CSMA/CA based Wireless Multihop Networks", *in Proc. of the IEEE International Symposium on on World of Wireless, Mobile and Multimedia Networks*, 2006, pp.143-149.

[2] C. Zhu and M. S. Corson, "QoS routing for mobile ad hoc networks", *in Proc. IEEE Infocom*, June 2001

[3] R. Min and A. Chandrakasan, "A framework for energy-scalable communication in high-density wireless networks",*in Proc. IEEE of ISLPED*, 2002, pp. 36-41.

[4] J.M. Miller, C. Sengul and I. Gupta, "Exploring the Energy-Latency Trade-Off for Broadcasts in Energy-Saving Sensor Networks", *in Proc. of IEEE ICDCS*, 2005, pp. 17-26.

[5] L. Y. Zhang, G. Ye and J. Hou, "Energy-efficient real-time scheduling in IEEE 802.11 wireless LANs", *ICDCS*, 2003, pp. 658-677

[6] Y. Yang, B. Krishnamachari and V.K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks", *in Proc. IEEE Infocom*, March 2004.

[7] R. Mangharam, S. Pollin, B. Bougard, R. Rajkumar, F. Catthoor, L. V. Perre and I. MOemann, "Optimal Fixed and Scalable Energy Management for Wireless Networks", *in Proc. of IEEE Infocom*, March 2005.

[8] S. Choi and K.G. Shin, "A Unified Wireless LAN Architecture for Real-Time and Non-Real-Time Communication Services", *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, 2000, pp.44-59.

[9] H. Li, P. Shenoy and K. Ramamritham, "Scheduling messages with deadlines in multi-hop real-time sensor networks", *in Proc. of IEEE RTAS*, May 2005, pp. 415-425.

[10] Q. Liu, S. Zhou, and G.B.Giannakis, "Cross-Layer Scheduling With Prescribed QoS Guarantees in Adaptive Wireless Networks", *IEEE Journal on Selected Areas in Communications*, vol. 23, No.5, May 2005.

[11] S. Pollin, R. Goyens, W. Cleeren and B. Bougard, "Cross-Layer Energy-Throughput Evaluation of Multi-hop/path Communication and Link Adaptation for IEEE 802.11a", *in Proc. of IEEE SIPS*, 2005.

[12] J. H. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-Hoc Networks", *in proc. of IEEE Infocomm*, 2000, pp. 22-31.

[13] K. Dudzinski and S. Walukiewicz, " Exact methods for the knapsack problem and its generalizations,", European Journal of Operations Reseach, vol. 28, 1987, pp. 3-21.

[14] E. 0. Elliott, "Estimates of error rates for codes on burst-noise channels", *Bell System Tech. J.*, vol. 42, Sept. 1963, pp. 1977-1997.

[15] C. Siva Ram Murthy and G. Manimaran, "Resource Management in Real-time Systems and Networks", *MIT Press, USA*, April 2001.