An Efficient Placement and Routing Technique for Fault-Tolerant Distributed Embedded Computing

ROOZBEH JAFARI and HASSAN GHASEMZADEH
University of Texas at Dallas
and
FOAD DABIRI, ANI NAHAPETIAN, and MAJID SARRAFZADEH
University of California, Los Angeles

This article presents an efficient technique for placement and routing of sensors/actuators and processing units in a grid network. The driver application that we present is a medical jacket, which requires an extremely high level of robustness and fault tolerance. The power consumption of such jacket is another key technological constraint. Our proposed interconnection network is a mesh of wires. A jacket made of fabric and wires would be susceptible to accidental damage via tears. By modeling the tears, we evaluate the probability of having failures on every segment of wires in our mesh interconnection network. Then, we study two problems of placement and routing in the sensor networks such that the fault tolerance is maximized while the power consumption is minimized. We develop efficient integer linear programming (ILP) formulations to address these problems and perform both placement and routing, simultaneously. This ensures that the solution is a lower bound for both problems. We evaluate the effectiveness of our proposed techniques on a variety of benchmarks.

Categories and Subject Descriptors: B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids—*Fault-tolerance and power optimization in grid sensor networks*; C.2.1 [**Network Architecture and Design**]: Circuit-Switching Networks; C.3 [**Special-Purpose and Application-Based Systems**]: Real-Time and Embedded Systems

General Terms: Algorithms, Management, Performance, Design, Reliability

Additional Key Words and Phrases: Distributed embedded system, fault tolerance, placement, routing, sensor networks

Authors' addresses: Roozbeh Jafari and Hassan Ghasemzadeh, Department of Electrical Engineering, Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, TX 75083; Foad Dabiri, Ani Nahapetian, and Majid Sarrafzadeh, Computer Science Department, University of California, Los Angeles, CA 90095; email: rjafari@utdallas.edu.

DOI 10.1145/1550987.1550991 http://doi.acm.org/10.1145/1550987.1550991

ACM Reference Format:
Jafari, R., Ghasemzadeh, H., Dabiri, F., Nahapetian, A., and Sarrafzadeh, M. 2009. An efficient placement and routing technique for fault-tolerant distributed embedded computing. ACM Trans. Embedd. Comput. Syst. 8, 4, Article 28 (July 2009), 26 pages.
DOI = 10.1145/1550987.1550991 http://doi.acm.org/10.1145/1550987.1550991

1. INTRODUCTION

1.1 Motivation

The past few years have seen exciting advances in the development of pervasive computing technologies. Computation, storage, and communication are now more or less woven into the fabrics of our society with much of the progress being due to the relentless march of Silicon-based electronics technology, as predicted by Moore's Law. The emerging field of flexible electronics, where electronic components such as transistors and wires are built on a thin flexible material, offers a similar opportunity to weave computation, storage, and communication into the fabric of the very clothing that we wear, thereby creating an intelligent fabric (also called electronic textiles or e-textiles) [Meoli and May-Plumlee 2002]. The implications of seamlessly integrating a large number of communicating computation and storage resources, mated with sensors and actuators, in close proximity to the human body are quite exciting; for example, one can imagine biomedical applications where biometric and ambient sensors are woven into the garment of a patient to trigger and modulate the delivery of a drug. Realizing such novel applications is not just a matter of developing innovative materials for flexible electronics, along with accompanying sensors and actuators. The characteristics of the flexible electronics technology and the requirements of the applications enabled by it necessitate radical innovation in system-level design. Electronic components built of flexible materials have characteristics that are very different from that of silicon and PCB-based electronics. Fault tolerance become more eminent in this architecture due to criticality of the applications running on this architecture. Furthermore, the operating scenarios of these systems involve environmental dynamics, physical coupling, resource constraints, infrastructure support, and robustness requirements that are distinct from those faced by traditional systems. This unique combination requires one to go beyond thinking of these systems as traditional electronic systems in a different form factor. Instead, a rethinking and complete overhaul of the system architecture and the design methodology for all layers of these systems is required.

1.2 Driver Application

Although the goal of our project is to investigate system architecture and design methodology concepts for electronic textiles, successful systems research of this nature requires that one not only develop these concepts, but also validate them in context of real technology constraints and application requirements. Therefore, we use pervasive patient monitoring and sensor-driven personalized

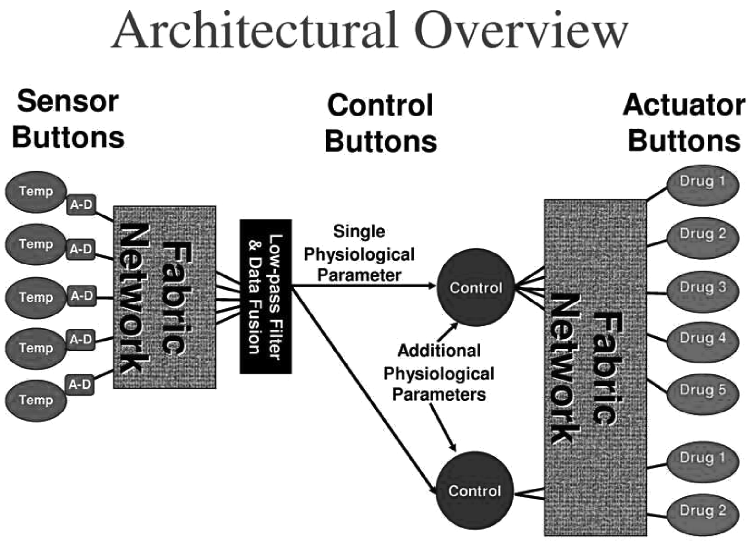

Fig. 1. System architecture of the medical jacket.

transdermal drug delivery as our driver application. One possibility of leveraging electronic textile technology in the context of such an application is to create a flexible garment (i.e., vest) that the patient can wear, which has sensing, computation, communication, and actuation elements embedded in it. We are developing such a prototype vest called medical jacket. Ideally, such a personalized drug delivery vest should have sensors on both the interior (to measure physiological readings) and on the exterior (to measure environmental readings, such as the presence of toxins in the surroundings), and a software-controlled, electrically actuated transdermal drug delivery system. It should allow low-latency, fine-grained adaptation of the drug dosage based on continual physiological measurements in the case of patients, and based on both environmental and physiological measurements in the case of people operating in hazardous environments. More generally, our application driver is representative of biomedical applications where information technology is integrated into fabrics and textiles. Figure 1 shows the overall system architecture of the medical jacket. As is evident from the Figure 1, the vest consists of four main subsystems: control (or computation), communication, sensing, and actuation.

2. RELATED WORK

Placement and routing problems on a grid have been studied extensively in the field of VLSI CAD. There has been considerable research effort to solve these problems. Like most other VLSI layout problems, it is believed that placement and routing cannot be solved optimally in a reasonable amount time [Sherwani 1995; Sarrafzadeh and Wong 1996]. Therefore, heuristic algorithms are used to obtain near-optimal solutions.

Placement algorithms can be generally classified into two main categories: constructive method and iterative improvement [Sarrafzadeh and Wong 1996]. Constructive methods generate a placement layout in a global sense. This category includes the min-cut algorithm [Dunlop and Kernighan 1985], the quadratic placement algorithm [Hall 1970], and the force-directed algorithm [Quinn and Breuer 1979]. Iterative improvement algorithms start from an initial solution and tries to improve the solution quality by making changes to the current solution. Simulated annealing [Sun and Sechen 1995] and genetic algorithm [Shahookar and Mazumder 1990] belong to the iterative improvement method. Regardless of the basic algorithm being used, most recent placement tools [Kleinhans et al. 1991; Caldwell et al. 1999; Wang et al. 2000; Chan et al. 2000] adopt hierarchical or multilevel paradigm by incorporating well-established graph partitioning [Karypis et al. 1997] or clustering techniques, in order to handle the high complexity of current designs.

The most basic approach to the routing problem is maze-running algorithm based on the breadth first search [Lee 1961]. Heuristic Steiner tree algorithms are used to find a minimum cost path for a multiterminal net. To route multiple nets, various approaches can be used such as sequential greedy algorithm, hierarchical approach [Burstein and Pelavin 1983], multicommodity flow-based algorithm [Shragowitz and Keel 1987], or integer linear programming [Lek and Kang 1992].

The main difference between our approach and other existing techniques is that we perform both placement and routing simultaneously. The solution that we find is also a lower bound for both problems. Our technique easily accommodates our requirements in terms of the instance size and the number of sensors, actuators, and processing units to be placed and routed.

Message routing in large interconnection networks has received a great deal of attention in recent years in the field of parallel computing. To decrease the amount of time in transmitting data, second generation multicomputers adopt wormhole routing mechanism, for example, Ametek 2010, nCUBE-2, Intel Paragon, J-machine, and iWARP [Ni and McKinley 1993]. Adaptive wormhole routing algorithms have been proposed for mesh, torus, and hypercube topologies. With wormhole routing, a message is divided into a sequence of fixed-size units of data, called flits. The header of a message contains all the information needed to decide the selection of next channel on the route. As the header flit advances, the remaining flits follow it in a pipeline fashion. When the header flit reaches a node that has no suitable output channel available, all of the flits in the message are blocked until the channel is freed. This form of routing technique makes the message transmission time almost independent of the distance between two nodes if the network is contention-free. A good survey article on wormhole routing techniques is done by Ni and McKinley [1993]. The main difference between our approach and wormhole routing is that we employ the concept of circuit switching for communication while wormhole routing utilized packet-switching technique. In our model, since we have continuous data collection from sensors, we employ the concept of circuit switching, which provides dedicated communication line between nodes. Barrenechea et al. [2004] studied optimal routing algorithms for regular sensor networks, namely, square and torus grid sensor networks, in both, the static case (no node failures) and the dynamic case (node failures). Their model, however, utilizes packet switching approach.

Multihop routing in wireless networks with respect to power optimization and fault-tolerance have been studied extensively. Generally speaking, a node in a wireless network independently explores its surrounding region and establishes connections with other neighboring nodes that are within its transmission and reception range. Our model significantly differs, since the interconnect network is not via wireless media. The wired interconnect networks reduces the environmental effects on the reliability of the network and greatly reduces the communication power consumption.

Furthermore, energy-aware routing for e-textile platforms was studied by Kao and Marculescu [2005]. They derive an analytical upper bound for the achievable number of jobs completed over all possible routing strategies. Their routing technique achieves about 50% of this analytical upper bound. Stanley-Marbell et al. [2003] addresses the modeling of computation, communication, and failure in e-textile and investigate the performance of code migration and remote execution. Despite their interesting approach, our main focus is to reconfigure/configure the architecture optimally based on routing/routing and placement. Furthermore, we model the tears and evaluate the probability of having failures on every segments of wires in our mesh interconnection network.

3. PRELIMINARIES

3.1 Interconnection Topology

The interconnection medium for our proposed system is a mesh of wire segments. The mesh interconnection topology is a wire-frame that has a regular structure, each vertex being connected to exactly four other vertices.

Mesh networks have several significant advantages. Each node has a dedicated communication link with every other node on the network and also has access to the full bandwidth available for that link. Nodes on buses must share the bandwidth available on the bus medium. Besides, in a mesh, multiple paths exist between devices. This brings a great robustness against faults. If a direct path between two nodes goes down, messages can be rerouted through other paths. Moreover, it has considerable scalability and can be easily manufactured. The manufacturing issues become more significant because in our system wires are integrated into the fabric, and this can be easily achieved with the current fabric manufacturing technology. Furthermore, the mesh interconnection is highly regular, which assists us in routing and placement of sensors and processing units. A picture of our fabric with mesh interconnection is shown in Figure 2.

In the remainder of the article, we refer to mesh interconnection topology by terminology “grid.” The interconnection topology is modeled as a grid graph illustrated in Figure 3. Each edge in the grid represent a wire or a communication link and each node represents a processing unit and a switch. Most nodes are

Fig. 2. Fabric with mesh of wires.

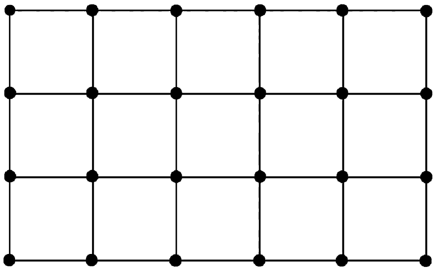

Fig. 3. Interconnection topology.

connected to four edges. Switches are responsible for either connecting the processing units to any of the edges or establish the connections between edges. In this article, we focus on routing and placement of sensors and processing units. We emphasize on finding disjoint paths between every pair of communicating nodes because our communication medium is a mesh of wires and each wire cannot handle more than one communication at a time. In addition, we chose the wired interconnection medium due to a high available bandwidth of wires and the requirements of our application (Electrocardiogram signals transmit at 115.2Kbps in our application).

3.2 Switch Model

Our proposed switch model is shown in Figure 4. Our switches are capable of connecting any terminals to any other terminals.

3.3 Circuit Switching vs. Packet Switching

Circuit switching has been considered for our proposed technique due to many advantages it offers to our applications. Our scheme has been designed for applications with high-bandwidth requirements. Therefore, dedicated paths for

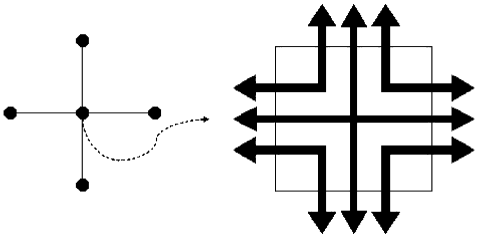

Fig. 4. Switch model.

communication enhances the overall performance of such systems. Moreover, due to low-delay requirements of most time-sensitive medical applications, the quality of service can be guaranteed through the circuit-switching approach. The circuit-switching scheme demands circuit setup time, which may assess an overhead on the system. Yet, once set up, its overhead on the network becomes negligible over time. In the packet-switching scheme, however, the path of each packet is determined independently. Each packet may travel by a different path. Considering our objective in finding the lower bound on the system lifetime and the complexity of our formulation, it may not be feasible to impose such overhead for every packet being transmitted. Finally, circuit-switching facilitates composition of larger packets and reduces the overhead of packet headers.

3.4 Power Lifetime

The mobility of our system requires the use of constrained power sources (i.e., batteries). Therefore, in the design of such system, power optimization is one of the major objectives. The amount of power resources available to the system determines the lifetime of the system, which will be referred to as power lifetime (l_p) in this article. As shown in the following equations, E indicates the remaining energy of the system power source, while P_w represents the power consumption rate of each node in the grid, which includes the power consumption of both the processing unit and the switch.

$$\begin{aligned} l_p &= \text{System power lifetime} \\ E &= \text{Energy remaining in the system power source} \\ P_w &= \text{power consumption of each active node} \end{aligned}$$

V is a set of all the nodes of the grid, while R_v consists of the nodes chosen and utilized for routing. Moreover, the size of set R_v is denoted by n.

$$V = \{v_i | v_i \text{ is a node in the grid}\}$$
$$R_v = \{v_i | v_i \in V \wedge v_i \text{ used in routing}\}$$
$$n = |R_v|$$

Finally, we define the power lifetime of the system as follows:

$$l_p = \frac{E}{P_w n} = \frac{1}{\lambda_p n} \text{ where } \lambda_p = \frac{P_w}{E}. \tag{1}$$

We also define λ_p, which will be used in the ILP problem formulation of routing and placement.

3.5 Power Consumption Model

In our proposed architecture, it is crucial to minimize the power consumption since the system is battery-powered. The power consumption due to processing is rather constant, as long as the processing units remain operational. We, however, can reduce the power consumption of internode communications. When a communication between a pair of nodes is established, a number of switches is utilized. The power consumption of the communication is proportional to the number of switches along the path considering that they must remain operational throughout the communication. The other important factor that is generally considered is the delay. However, in our system, the communications are established using the circuit switching scheme. Therefore, the wire length of each path is rather small and delay concerns become a nonissue.

3.6 Fault Lifetime

As discussed in Section 3.1, our system is made of fabric, which is susceptible to accidental damage via tears and punctures. Such faults may yield loss of a connection and, therefore, may result in total system failure. Firefighters, policemen, soldiers, astronauts, and athletes working in hazardous environments may experience more faults than others. We define λ_f as the rate of fault occurrence per unit time. We define E and R_e as set of edges in the grid and set of edges used in routing, respectively:

$$\begin{aligned} E &= \{e_i | e_i \text{ is an edge in the grid}\} \\ R_e &= \{e_i | e_i \in E \wedge e_i \text{ used in routing}\}. \end{aligned}$$

The set of edges not used in routing is defined by R_e':

$$\begin{aligned} R_e' &= E - R_e \\ &= \{e_i | e_i \in E \wedge e_i \text{ NOT used in routing}\}. \end{aligned}$$

When a fault occurs, probability of loosing a particular edge e_i is represented by $P(e_i)$.

$$P(e_i) = \Pr\{e_i \text{ fails} | \text{a fault has occurred}\} \tag{2}$$

Assuming that the routing is performed (i.e., the edges for intercommunication between nodes are selected), when a fault happens, the probability of having system failure due to link failures is depicted by $P(e_R)$.

$$\begin{aligned} P(e_R) &= \Pr\{\text{an edge in } R_e \text{ fails} | \text{a fault has occurred}\} \\ &= \textstyle\sum_{e_i \in R_e} P(e_i) = \alpha \end{aligned}$$

Obviously, failures in $e_i \in R_e'$ does not affect the system functionality, since $e_i \in R_e'$ is not utilized for routing. The probability of not having system failure when a fault occurs is shown with $P(e_{R'})$.

$$P(e_{R'}) = 1 - P(e_R) = 1 - \alpha$$

The expected fault lifetime of the system now can be defined in the following equation. The first term in the series indicates the probability of having system failure after the occurrence of the first fault. The second term denotes the probability of system failure with two faults, while the first fault refrains the system intact. The same arguments hold for all the remaining terms.

$$\begin{aligned}\overline{l_f} &= \text{Expected lifetime of the system}\\ &= \frac{\alpha}{\lambda_f} + \frac{2(1-\alpha)\alpha}{\lambda_f} + \frac{3(1-\alpha)^2\alpha}{\lambda_f} + \cdots\\ &= \frac{\alpha}{\lambda_f}\sum_{i=1}^{\infty} i(1-\alpha)^{i-1}\end{aligned}$$

The expected fault lifetime is shown by Eq. (3) where λ_f and α are the fault occurrence rate and probability of having system failure, respectively.

$$\overline{l_f} = \frac{\alpha}{\lambda_f(1-(1-\alpha))^2} = \frac{1}{\lambda_f \alpha} \tag{3}$$

3.7 Fault Model

In this section, we model “tears” or “cuts” that may occur to the fabric and yield loss of a connection in the network. For simplicity, we consider only monotone tears. In practice, this assumption is not far from reality because the results from tear and puncture analysis experiments on textile confirm that the nature of tears is usually monotone (WARP and WEFT tears) [Witkowska and Frydrych 2004]. Various mechanical instruments are being used for these experiments such as Elmendorf (falling pendulum) [Bona 1994] and COMPUTE (COMbined PUncture and TEar) [Primentas 2001] on cotton fabric. In addition, several standards have been implemented for tear analysis on textile such as PE-P-04640:1976, “Fabric measurement methods. Woven and knitted fabrics. Determination of tear strength” and PE-EN ISO 13937-3:2002:1-4 “Textiles—Tear properties of fabrics—Parts 1–4.”

Unfortunately, most reports published in this area only describe statistics about the strength/length of tears and not about their physical distribution throughout the fabric. Therefore, the following theoretical tear analysis was carried out with the assumptions based on the outcome of practical tear analysis experiments.

In Figure 5, a few tears are shown. Each tear can be modeled as a path between two nodes in the dual graph of the grid, as shown in Figure 6. The dual graph refers to the graph in which each region of the grid is presented with a node. Whenever two regions in the grid have an edge in common, there exists an edge between corresponding nodes in the dual graph. According to this model, the total number of different tears that may happen in the fabric is the total number of distinct paths in the dual graph. Therefore, the number of distinct tear occurrence on a wire segment in the grid is the same as the number of paths going through the corresponding edge in the dual graph. Consequently, we must count the number of paths that pass through each edge in the dual graph. To show how this is performed, initially, we present the following lemmas.

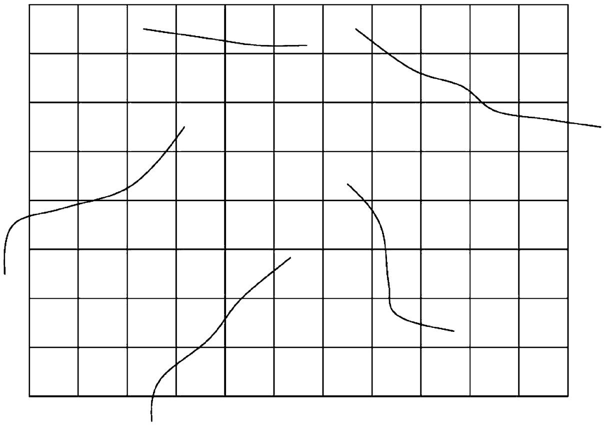

Fig. 5. Tear model.

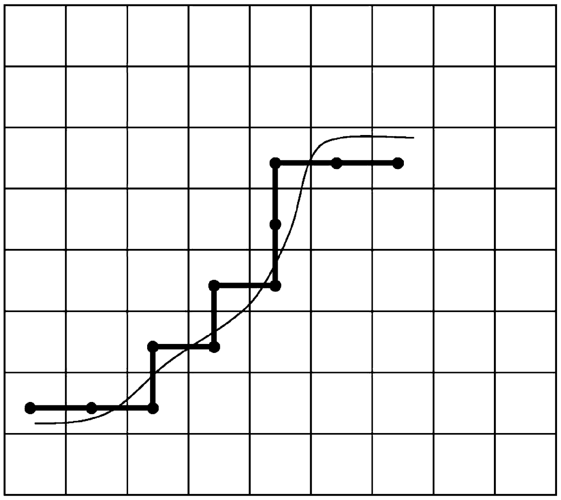

Fig. 6. Representation of tears in dual graph.

LEMMA 3.1. *The number of distinct paths from node v_{11} to v_{nm} on a grid is:*

$$P(v_{11}, v_{nm}) = \binom{m+n-2}{n-1}. \tag{4}$$

PROOF. Length of each path from v_{11} to v_{nm} is $m+n-1$ and each path can be represented with a string of length $m+n-2$ containing of right (R) and up (U) moves. Each string has exactly $n-1$ (U)'s. The total number of distinct strings is shown in Equation (4), which is equivalent to the number of paths. □

Now, consider an edge w which connects nodes v_{ij} to v_{ij+1}, as shown in Figure 7. Since we consider only monotone tears, the total number of paths

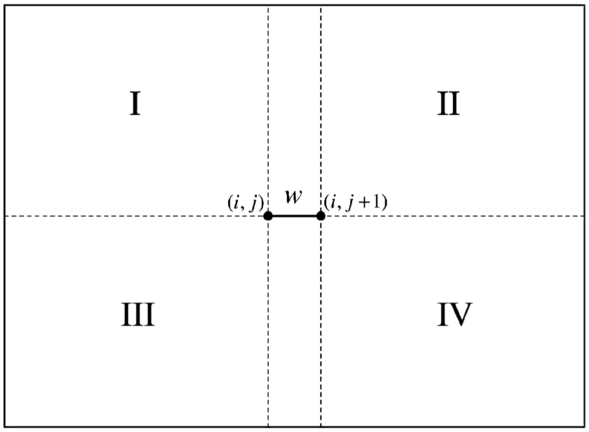

Fig. 7. Region subdivisions for every edge in dual graph.

that pass through w, $T(w)$, can be evaluated by Eq. (5):

$$\begin{aligned} T(w) \;=\; & R(I, v_{ij}) \times R(IV, v_{ij+1}) \\ & + R(III, v_{ij}) \times R(II, v_{ij+1}). \end{aligned} \tag{5}$$

$R(X, v_{ij})$ is the total number of paths connecting every node in rectangle X to node v_{ij}. Possible rectangles for X are shown in Figure 7. We have:

$$R(I, v_{ij}) = \sum_{s=1}^{i} \sum_{t=1}^{j} P(v_{st}, v_{ij}).$$

To find a closed form for $R(I, v_{ij})$, we observe that:

$$\sum_{s=1}^{i} P(v_{s1}, v_{ij}) = \sum_{s=1}^{i} \binom{j-1+i-s}{j-1} = \binom{j+i-1}{j}.$$

This implies that the total number of paths originating from every node on the left border of rectangle I to v_{ij} is equal to $P(v_{11}, v_{ij+1})$. Finally, we conclude that:

$$R(I, v_{ij}) = \sum_{t=1}^{j} \binom{i+t-1}{t} = \binom{i+j}{j} - 1. \tag{6}$$

Similar equations hold for $R(IV, v_{ij+1})$, $R(III, v_{ij})$, $R(II, v_{ij+1})$. Note that we only considered horizontal edges. The same argument is valid for vertical edges. We previously showed that each tear in the grid is represented by a path in the corresponding dual graph. Therefore, assuming that a tear occurs, the probability of loosing an edge e_i in the grid is proportional to the number of distinct paths in the dual graph crossing that edge. In other words, it would be proportional to the total number of paths that passes through w_i (i.e., $T(w_i)$), which corresponds to e_i in the dual graph. To find the probability of loosing an edge, we calculate the total possible tears in the network, which is equivalent to all possible paths between any two nodes (v_{ij}, v_{pq}) in the dual graph. Therefore, the total number of possible tears in the network is:

$$T_N = 2 * \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} \sum_{p=1}^{i} \sum_{q=1}^{j} \binom{j-q+i-p}{j-q} - T_{straight}. \tag{7}$$

The nested summation is taken over all pairs of (v_{ij}, v_{pq}) and is multiplied by two to take into account both descending and ascending paths. $T_{straight}$ is the total number of straight paths including paths with length zero (single nodes). Since straight paths are being considered as both ascending and descending, they will be counted twice in the first term of Eq. (7). Therefore, $T_{straight}$ must be deducted from the nested summation in Eq. (7). $T_{straight}$ can be simply written as:

$$T_{straight} = (n+1)\binom{m+1}{2} + (m+1)\binom{n+1}{2} + (n+1)(m+1). \tag{8}$$

The latter term in Eq. (8) is basically the number of paths in the graph with length zero (the number of nodes in the graph). Therefore, the total number of possible tears in the network can be expressed as:

$$\begin{aligned} T_N \;=\; & 2 * \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} \left(\binom{i+j}{j} - 1 \right) - [(n+1)\binom{m+1}{2} + (m+1)\binom{n+1}{2} \\ & + (n+1)(m+1)]. \end{aligned} \tag{9}$$

Hence, probability of having failure on edge e_k is calculated as follows:

$$Pr(e_k) = \frac{T(w_k)}{T_N}. \tag{10}$$

In Eq. (10), each w_i in dual graph corresponds to e_i in the grid. Figure 8 illustrates the probability of failure on each edge in the grid when a tear occurs.

In the model we illustrated, our analysis show that some wires are more susceptible to failures. If the wires already utilized for communication fail, the system will fail. Furthermore, since the lifetime of the system is determined by the shortest lifetime of all wire segments selected for routing, we prefer to employ wires, which are less susceptible to faults.

3.8 Integer Linear Programming Formulation

ILP is used to specify problems where a function has to be maximized or minimized, and the variables are constrained by inequality and equality constraints and/or integral restrictions on either a subset or all the variables. The objective function and the inequality or equality constraints are all linear. We define the mixed integer programming problem (MIP) as:

$$max\{cx + hx : Ax + Gy \le b, x \in Z_+^n,\, y \in R_+^n\},$$

where Z_+^n is the set of nonnegative integral n-dimensional vectors, R_+^n is the set of nonnegative integral p-dimensional vectors, and $x = \{x_1, \ldots, x_n\}$ and $y = \{y_1, \ldots, y_p\}$ are the variables in the problem. We restrict our attention to

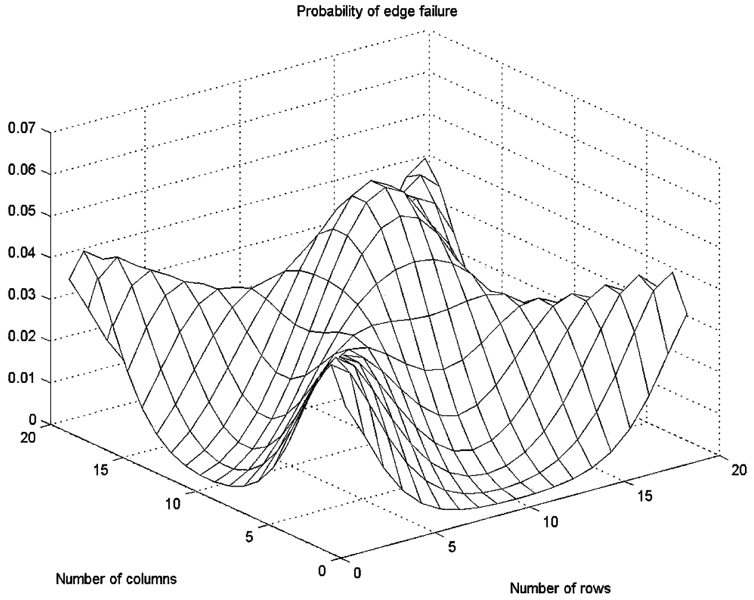

Fig. 8. Probability of edge failure in the grid for all possible distinct tears.

a special case of the MIP where all variables may have only integer values. The integer linear programming (ILP) is defined as

$$max\{cx : Ax \leq b, x \in Z_{+}^{n}\}.$$

More specifically, we restrict the integer values of x to be either 0 or 1. Therefore, 0-1 ILP where x is redefined as $x \in B_n$, where B_n is a set of n-dimensional binary vector.

3.9 Terminology

In the remainder of this article, we will specify the ILP problem in the following way.

$$y = MAX(cx) \tag{11}$$

$$Ax \geq b \tag{12}$$

Eq. 11 specifies the objective function for the problem where we define $x = x_1, \ldots, x_n$ as 0-1 variables. The constraints of the problem can be formulated in matrix form, as shown in Eq. (12). In the remainder of the article, we will use uppercase letter representation to specify constant matrices or values, and lowercase letters to represent variables.

4. PROBLEM DEFINITION

We introduced the terminologies power lifetime and fault lifetime as well as the tear analysis technique. The lifetime of our system is highly dependent on the remaining energy in the power source and the frequency of fault occurrence.

Sometimes, the system may fail due to lack of energy and in some cases, faults may parallelize the system. To increase the power lifetime, once we perform placement and routing, we seek to minimize the total length of routing. To improve the fault lifetime, we attempt to use the edges, which are less susceptible to faults. The overall lifetime of the system is dependent both on power and fault lifetime. Therefore, to maximize the overall lifetime of the system, we shall maximize the minimum of power and fault lifetimes.

$$MAX(MIN(l_p, \overline{l_f})) \tag{13}$$

However, the objective function in an ILP formulation may not contain both *MAX* and *MIN*, as shown in Eq. (13). Therefore, we redefine the objective functions as illustrated in Eqs. (14) and (15) and have the solver optimize both objectives individually in two different runs.

$$\begin{gathered} MAX(l_p) \\ \text{with constraint } l_p < \overline{l_f} \end{gathered} \tag{14}$$

$$\begin{gathered} MAX(\overline{l_f}) \\ \text{with constraint } \overline{l_f} < l_p \end{gathered} \tag{15}$$

The maximum value of Eqs. (14) and (15) is the value of Eq. (13). In Section 5, we specify in details how the objective functions are constructed.

5. PROBLEM FORMULATION

5.1 Routing

In both placement and routing problems, we assume that the interconnection network is a grid of size $m \times n$. The vertices correspond to processing units along with switches, while the edges correspond to the links between them. Each vertex and edge in the grid is assigned a variable. These variables specify if an edge and/or vertex is selected for routing and/or placement. Further details on how these variables are used will be explained later. However, the accordance between the indices of variables and the position of vertices and/or edges are as follows.

Variable x_{pij} represent if node with coordinations i and j in the grid is chosen for the pth communication pair. As shown in Figure 9, variables y_{hij} and y_{vij} represent horizontal and vertical edges corresponding to node ij in the grid, respectively.

Problem: *Routing to Minimize the Power Consumption*
Instance: *Graph $G(V, E)$ s.t. G is a grid, collection of disjoint vertex pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_l, t_k)$*
Question: *Find k mutually vertex-disjoint paths in G, connecting s_i to t_i for each i, $1 \leq i \leq k$ s.t. the total length of paths is minimum*

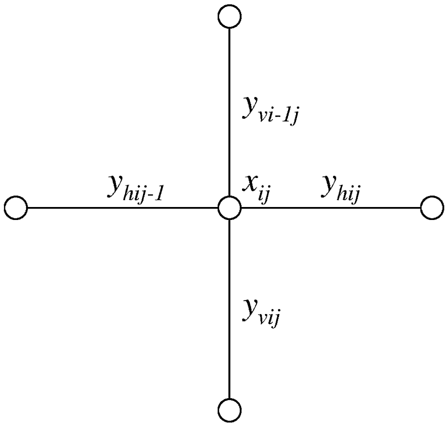

Fig. 9. ILP variables.

In the ILP formulation, we define variables x_{pij} where $p = 1, \ldots, k$. x_{pij} represents whether vertex v_{ij} is selected as a part of the path between (s_p, t_p). We also define q_{pij} as follows:

$$x_{pij} = \begin{cases} 1, & \text{if vetex } v_{ij} \text{ is selected for} \\ & \text{communicating pair } (s_p, t_p) \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

$$y_{(h/v)pij} = \begin{cases} 1, & \text{if edge } y_{(h/v)pij} \text{ is selected for} \\ & \text{communicating pair } (s_p, t_p) \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

$$q_{pij} = \begin{cases} 1, & \text{if vertex } v_{ij} = s_p \vee v_{ij} = t_p \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Using these variables, we define the constraints in Eqs. (19), (20), and (21) for the ILP formulation. Eq. (19) ensures that a feasible path is found between s_p and t_p while Eq. (20) denotes that an edge in the grid may not be used for more than one path. Eq. (21) enforces that the paths are vertex disjoint. To ensure that a path is feasible, we enforce the condition that exactly two neighboring edges of a vertex must be selected if the vertex is chosen for routing. In the case where the vertex is not chosen, none of the neighboring edges may be selected. This condition is slightly different for the source and sink vertices of the path, since exactly one neighboring edge must be selected for routing. This condition is expressed in Eq. (19).

Finally, the objective function, represented in Eq. (22), attempts to minimize the power consumption or the number of nodes employed for routing. Subsequently, this would maximize the power lifetime.

$$\begin{gathered} 2x_{pij} - y_{phij-1} - y_{pvi-1j} - y_{phij} - y_{pvij} - q_{pi} = 0 \\ 1 \leq p \leq k \\ 1 \leq i \leq M \\ 1 \leq j \leq N \end{gathered} \tag{19}$$

$$\sum_{p=1}^{k} y_{p(h/v)ij} \leq 1 \tag{20}$$

$$\sum_{p=1}^{k} x_{pij} \leq 1 \tag{21}$$

$$MIN\left(\sum_{p=1}^{k}\sum_{i=1}^{m}\sum_{j=1}^{n} x_{pij}\right) \tag{22}$$

Problem: *Routing to Maximize the Fault-Tolerance*
Instance: *Graph $G(V, E)$ s.t. G is a grid, fault occurrence weights on edges of G, collection of disjoint vertex pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_l, t_k)$*
Question: *Find k mutually vertex-disjoint paths in G, connecting s_i to t_i for each i, $1 \leq i \leq k$ s.t. (23) is minimum.*

Tear analysis described in Section 3.7 generates probability of having failures in every edge of the grid once a tear occurs. This information helps us to improve the fault tolerance of the system. As opposed to the previous problem, the objective function is to pick edges for routing such that the probability of fault occurrence in the selected edges is minimized. Therefore, the constraints in Eqs. (19), (20), and (21) still hold while the objective function in Eq. (22) changes according to the new requirements.

$$MIN\left(\sum_{p=1}^{k}\sum_{i=1}^{m}\sum_{j=1}^{n}(P(e_{hij})y_{hpij} + P(e_{vij})y_{vpij})\right) \tag{23}$$

Problem: *Routing to Maximize the Fault Tolerance and Minimize the Power Consumption*
Instance: *Graph $G(V, E)$ s.t. G is a grid, fault occurrence weights on edges of G, collection of disjoint vertex pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_l, t_k)$*
Question: *Find k mutually vertex-disjoint paths in G, connecting s_i to t_i for each i, $1 \leq i \leq k$ s.t. (25) and/or (25) is minimum.*

The first problem that we discussed earlier in Section 5.1 addresses the issue of power optimization. It attempts to minimize the total number of nodes selected for routing. Since selecting each node in the grid requires activation of a switch and a processing unit, the objective function minimizes the power consumption of the network. The previous problem, however, denotes how we can improve the fault tolerance of our interconnection medium through using edges, which are less exposed to faults. Now, we present a new problem formulation, which addresses both power optimization and fault-tolerance issues simultaneously. Eqs. (19), (20), and (21) are still valid; however, Eqs. (24) and (25) determine the lower bound for the lifetime of our system, as described in Section 4.

$$MIN\left(\sum_{p=1}^{k}\sum_{i=1}^{m}\sum_{j=1}^{n}x_{pij}\right)$$
$$\text{with respect to}$$
$$\sum_{p=1}^{k}\sum_{i=1}^{m}\sum_{j=1}^{n}x_{pij} > \tag{24}$$
$$\sum_{p=1}^{k}\sum_{i=1}^{m}\sum_{j=1}^{n}(P(e_{hij})y_{hpij}+P(e_{vij})y_{vpij})$$

$$MIN\left(\sum_{p=1}^{k}\sum_{i=1}^{m}\sum_{j=1}^{n}(P(e_{hij})y_{hpij}+P(e_{vij})y_{vpij})\right)$$
$$\text{with respect to}$$
$$\sum_{p=1}^{k}\sum_{i=1}^{m}\sum_{j=1}^{n}(P(e_{hij})y_{hpij}+P(e_{vij})y_{vpij}) > \tag{25}$$
$$\sum_{p=1}^{k}\sum_{i=1}^{m}\sum_{j=1}^{n}x_{pij}$$

The next problem that we present is an extension to the previous problem where each edge in the grid carries more than one wire. We define the channel capacity for each edge in the grid and assume that each edge contains c wires. Thus, at most, c disjoint paths may use one edge simultaneously.

Problem: *Routing to Maximize the Fault Tolerance and Minimize the Power Consumption*
Instance: *Graph $G(V, E)$ s.t. G is a grid, fault occurrence weights on edges of G, collection of disjoint vertex pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_l, t_k)$*
Question: *Find k mutually vertex-disjoint paths in G, connecting s_i to t_i for each i, $1 \leq i \leq k$ s.t. each edge in E is not associated with more than c paths and (25) and/or (25) is minimum.*

With our new assumption, the constraint in Eq. (19) is still effective, while Eqs. (26) and (27) are enforced as well. The objective function remains the same as Eqs. (24) and (25).

$$\sum_{p=1}^{k} y_{p(h/v)ij} \leq c \tag{26}$$

$$\sum_{p=1}^{k} x_{pij} \leq c \tag{27}$$

The last ILP formulation for routing is for the case where a processing unit is connected to multiple sensors. This case happens where multiple sensors communicate with a single processing unit simultaneously and efficiently.

Problem: *Routing to Maximize the Fault Tolerance and Minimize the Power Consumption*
Instance: *Graph $G(V, E)$ s.t. G is a grid, fault occurrence weights on edges of G, collection of disjoint tuples $(S_1, t_1), (S_2, t_2), \ldots, (S_k, t_k)$*
Question: *Find k mutually vertex-disjoint paths in G, connecting all $s_i \in S_i$ to t_i for each i, $1 \leq i \leq k$ s.t. each edge in E is not associated with more than c paths and (24) and/or (25) is minimum.*

Eqs. (20), (21), and (26) still hold. Only Eq. (19) is modified and redefined in Eq. (28). $|S_p|$ is the cardinality (i.e., the number of members) of S_p.

$$\begin{gathered} |S_p| x_{pij} - y_{phij-1} - y_{pvi-1j} - y_{phij} - y_{pvij} - q_{pi} = 0 \\ 1 \leq p \leq k \\ 1 \leq i \leq M \\ 1 \leq j \leq N \end{gathered} \tag{28}$$

5.2 Placement

Traditional placement and routing problems in VLSI CAD are considered as two separate problems. In this section, we present an ILP formulation where we solve both placement and routing of our sensor network simultaneously. In most cases, the medical application itself derives the location of sensors. However, in some applications, the flexibility of placing sensors within a specific range may be bestowed. This can be easily accommodated by our proposed ILP technique and the optimal location of sensor to minimize our objective function will be found. For instance, a temperature sensor may be required to be placed on chest, but its exact location is not given. In this case, our technique attempts to optimize its location too.

Problem: *Placement and Routing to Maximize the Fault Tolerance and Minimize the Power Consumption*
Instance: *Graph $G(V, E)$ s.t. G is a grid, fault occurrence weights on edges of G, collection of disjoint vertex set $(S_1, S'_1), (S_2, S'_2), \ldots, (S_k, S'_k)$* s.t. $s_i \in S_i \subseteq V$ and $t_i \in S'_i \subseteq V$, S_i and S'_i are set of nodes which can potentially be source or sink of a path.
Question: *Find k mutually vertex-disjoint paths in G, connecting s_i to t_i for every i, $1 \leq i \leq k$ s.t. (24) and/or (25) is minimum.*

In this problem, the nodes that are candidates to be either source or sink of the paths are given in sets S_i's and $S_i^{\prime}$'s. For each tuple $(S_i, S'i)$, however, exactly one node from S_i and one node from S'_i will be selected. In the case where the position of a node is fixed (i.e., the node must be placed at a specific position), the set S_i or S'_i may have only one member. To solve this problem, our ILP formulation is modified, and we define two sets of new variables for each

Table I. Power/Fault Lifetime Comparison in 5×5 Grid

	Fault Lifetime (hours)	Power Lifetime (hours)
$\lambda_f = 1/24\,h^{-1}$	51.35	N/A
$\lambda_f = 1/30\,h^{-1}$	56.25	55.30
$\lambda_f = 1/36\,h^{-1}$	N/A	64.38

communication pair.

$$\begin{gathered}\textstyle\sum_{s_{pij}\in S_p} s_{pij} = 1\\ 1 \le p \le k\\ 1 \le i \le M\\ 1 \le j \le N\end{gathered} \tag{29}$$

$$\begin{gathered}\textstyle\sum_{s'_{pij}\in S'_p} s'_{pij} = 1\\ 1 \le p \le k\\ 1 \le i \le M\\ 1 \le j \le N\end{gathered} \tag{30}$$

$$\begin{gathered}2x_{pij} - y_{phij-1} - y_{pvi-1j} - y_{phij} - y_{pvij} - s_{pij} - s'_{pij} = 0\\ 1 \le p \le k\\ 1 \le i \le M\\ 1 \le j \le N\end{gathered} \tag{31}$$

The constraints in Eqs. (29) and (30) enforce that only one node from sets S_i or S'_i is selected as a source or a sink of a path. Eq. (31) is modified from Eq. (19) to ensure that a feasible path for every communication pair is found. The constraints in Eqs. (20), (21) and the objective function in Eqs. (24) and (25) still hold for this problem.

We can apply all the variations of routing problems described in Section 5.1 to the placement and routing formulations.

6. EXPERIMENTAL ANALYSIS

In this section, we present the simulation results on fault/power lifetime optimization on various benchmarks. We used the CPLEX solver on a 400MHz Sun Ultra-10 machine and allowed the maximal runtime to be 30 minutes. Various benchmarks are generated for the experimental analysis. The size of benchmarks varies from 5 × 5 to 11 × 11. The number of communication pairs are altered as well. The communication pairs are placed across the grid randomly for each instance size. The power consumption of the processing units is assumed to be $P_w = 24mW$ (Berkeley mica2dots—power supply voltage $\cong 3V$). We also assume that the system is equipped with ten $1.5V$, $2000mAh$ AA batteries. Hence, $\lambda_p = 0.0008h^{-1}$. Fault occurrence rate, λ_f, picks the values of $1/24h^{-1}$, $1/30h^{-1}$, and $1/36h^{-1}$. $\lambda_f = 1/24h^{-1}$ implies that, on average, a random tear occurs in every 24 hours.

Table I illustrates the simulation results for a 5×5 grid with 8 communication pairs. The first and the second columns present the values of Eqs. (25) and (24),

Table II. Power/Fault Lifetime Comparison in 7×7 Grid

	Fault Lifetime (hours)	Power Lifetime (hours)
$\lambda_f = 1/24\, h^{-1}$	31.74	N/A
$\lambda_f = 1/30\, h^{-1}$	33.71	33.15
$\lambda_f = 1/36\, h^{-1}$	N/A	40.11

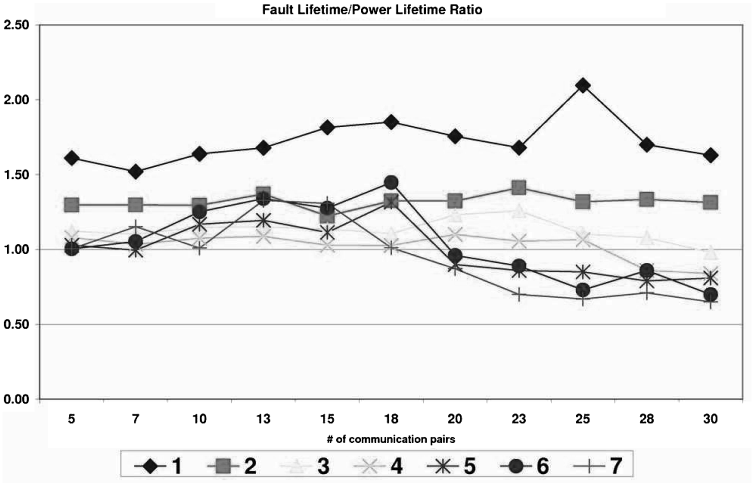

Fig. 10. Fault/power lifetime ratio for 7×7 mesh size.

respectively. As shown in the Table I, for small values of λ_f (i.e., = 1/24), the routing must be performed with most emphasis on power lifetime optimization while for larger λ_f's, (i.e., = 1/36), the fault lifetime is optimized to achieve the best results. For $\lambda_f = 1/30h^{-1}$, both fault and power-objective functions yields approximately similar lifetimes. Table II exhibits similar simulation results for a 7×7 grid. Figure 10 represents fault/power lifetime ratio for a 7×7 grid. In this set of experiments, the location of source and sink nodes in each communication pair was selected randomly across the grid such that the rectilinear distance between source and sink nodes is not greater than l_{max}, where $1 \leq l_{max} \leq 7$. Both the number of communication pairs and l_{max} are varied. Each plot in Figure 10 corresponds to a specific l_{max}. For each data point, 20 benchmarks were generated and the average ratio of all 20 benchmarks is shown in the diagram. Overall, 1,540 benchmarks were generated for $\lambda_f = 1/30h^{-1}$. A ratio greater than 1 indicates that the system is less sensitive to faults and most probably dies due to the power outage while ratio less than 1 implies that most likely the system becomes paralyzed due to the faults. According to Figure 10, the fault/power lifetime ratio drops to less than 1 for large number of communication pairs distributed randomly with $l_{max} \geq 3$. Hence, more efforts must be spent on fault lifetime optimization of such benchmarks while in less congested networks, power lifetime gains more importance. This is due to the fact that in congested networks, we are forced to use edges that are highly susceptible to faults and therefore fault lifetime decreases rapidly while the power lifetime is not decreasing at the same order.

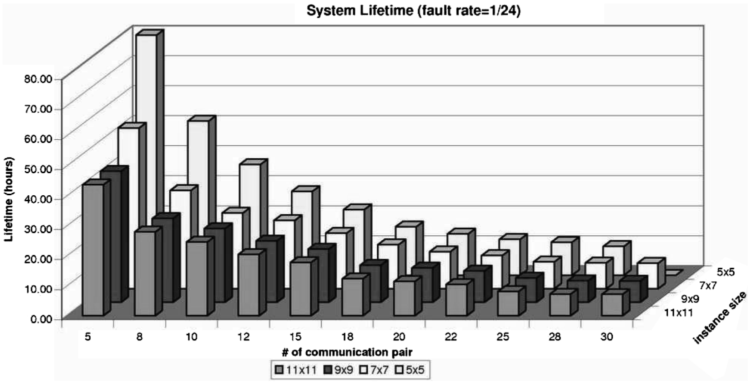

Fig. 11. Lifetime analysis ($\lambda_f = 1/24\, h^{-1}$).

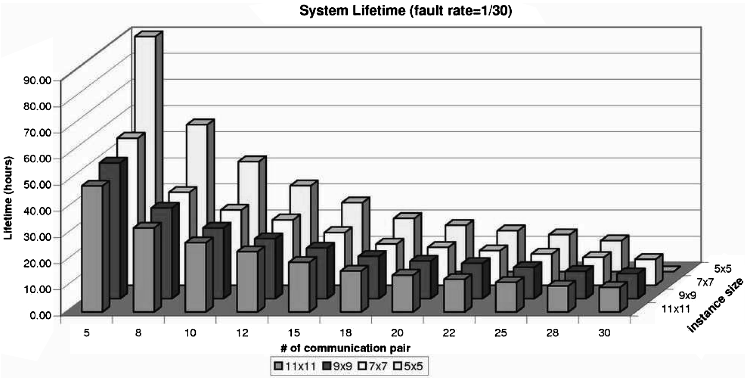

Fig. 12. Lifetime analysis ($\lambda_f = 1/30\, h^{-1}$).

Figures 11, 12, and 13 present lower bounds for both power and fault lifetime of systems with various configurations. Each data point again reflects the average of lifetime acquired from 20 benchmarks. In each benchmark, the source and the sink of communication pairs are placed with uniform distribution across the grid, where l_{max} = instance size. As shown, once the fault occurrence rate, λ_f, increases, the overall lifetime of the system decreases and more efforts must be spent on fault tolerant routing. As illustrated in all diagrams, the solver was unable to find a feasible solution for any 30 communication pairs placed randomly on a 5 × 5 grid within 30 minutes.

In Figures 14, 15 and 16, the lower bounds for both power and fault lifetime are represented where the size of the grid was set to 7 × 7 and l_{max} as well as the number of communication pairs are varied accordingly. Each data point is the average of 20 benchmarks. The other observation is that when the number of communication pairs grows, if the network becomes relatively congested, both power and fault lifetime become relatively constant and similar. The reason is that most resources of the system are already utilized and the routing algorithm do not have the flexibility of choosing the best solution with respect to either power optimized or fault-tolerant routing techniques.

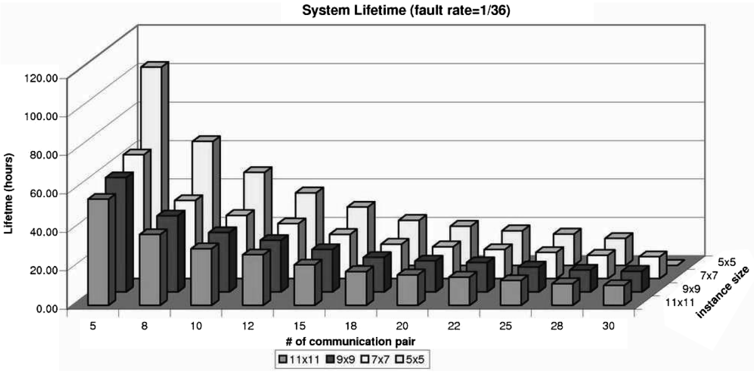

Fig. 13. Lifetime analysis ($\lambda_f = 1/36\, h^{-1}$).

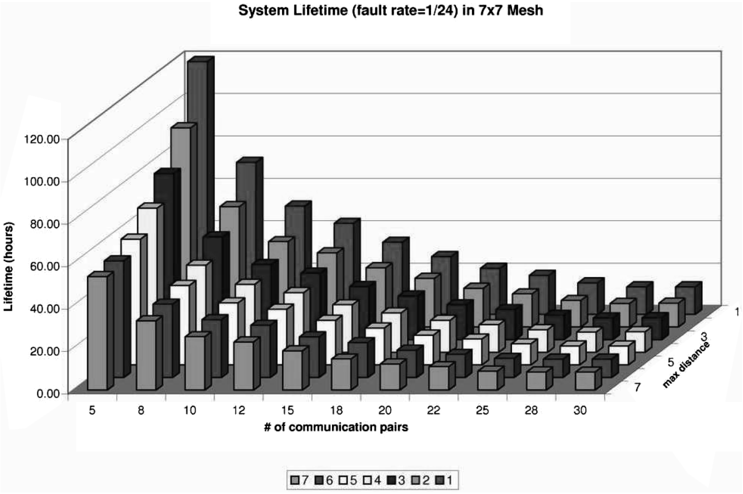

Fig. 14. Lifetime analysis on fixed mesh-size varying communication distance($\lambda_f = 1/24\, h^{-1}$).

Our objective may not be achieved by a greedy method since it attempts to maximize the minimum of two objective functions. However, to exhibit the benefits of our proposed formulation over a greedy approach, we considered a method where a communication pair is selected randomly and its shortest path based on the power objective function is calculated. This process is repeated until shortest paths between all pairs are determined. If a feasible solution is not found, another sequence of communication pairs is selected until a solution is found. The system lifetime in such network corresponds to the minimum of fault and power lifetimes. The same greedy approach is repeated while the fault objective functions is considered. The greedy fault lifetime is compared against the greedy power lifetime and the maximum is picked and reported in Figures 17 and 18. These set of experiments are performed on a 7 × 7 grid, while the fault rate are assumed to be $\lambda_f = 1/24h^{-1}$ and $\lambda_f = 1/36h^{-1}$ for Figures 17 and 18, respectively. Each data point is the average of 20×7 random benchmarks where $1 \leq l_{max} \leq 7$. In some benchmarks, the greedy approach was

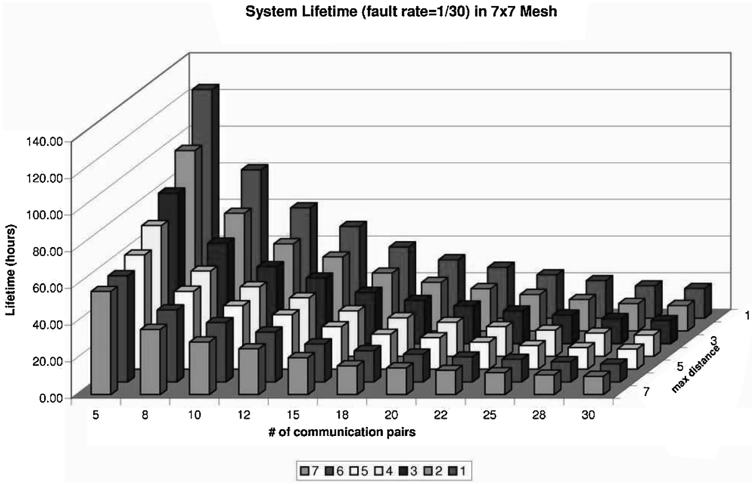

Fig. 15. Lifetime analysis on fixed mesh-size varying communication distance($\lambda_f = 1/30\,h^{-1}$).

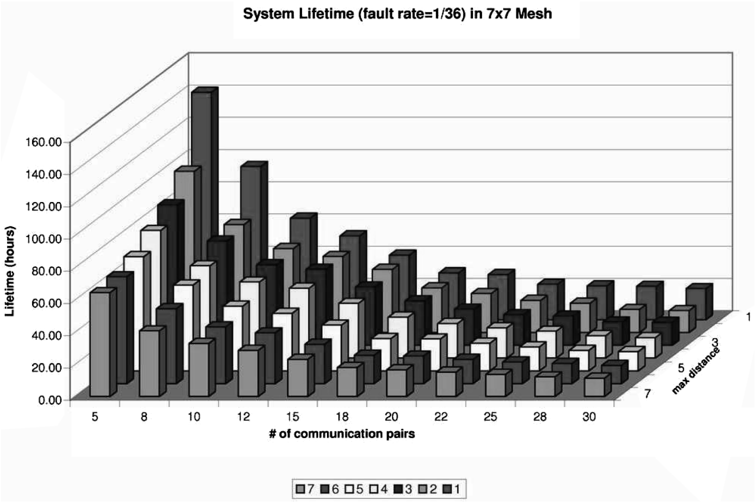

Fig. 16. Lifetime analysis on fixed mesh-size varying communication distance($\lambda_f = 1/36\,h^{-1}$).

not successful in finding a feasible solution. In such cases, the benchmark did not contribute to the data point. On average, the lifetime of the system designed by our proposed technique for $\lambda_f = 1/24h^{-1}$ is 79% higher than the greedy method. The improvement has been 103% for $\lambda_f = 1/36h^{-1}$. It illustrates the effectiveness of our proposed technique when the fault rate increases.

7. CONCLUSION AND FUTURE WORK

Our proposed ILP technique has the two following properties:

—It considers placement and routing problems at the same time targeting our performance objective. Generally, placement and routing are being treated

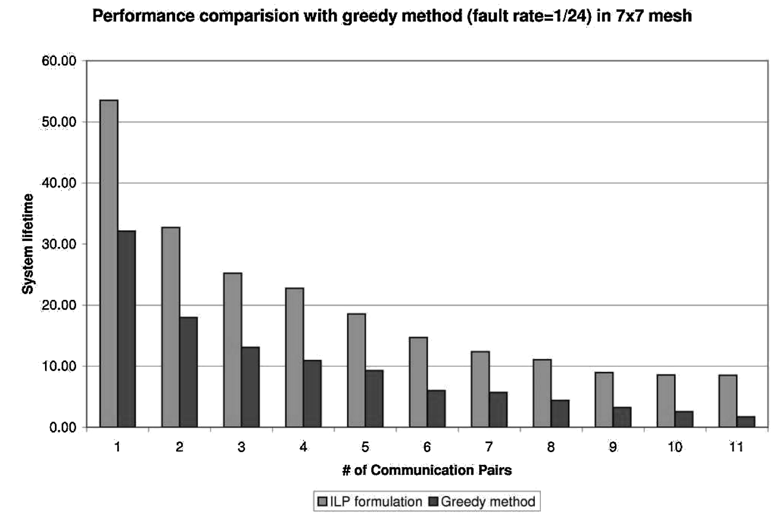

Fig. 17. Performance comparison of our technique with a greedy method ($\lambda_f = 1/24\, h^{-1}$).

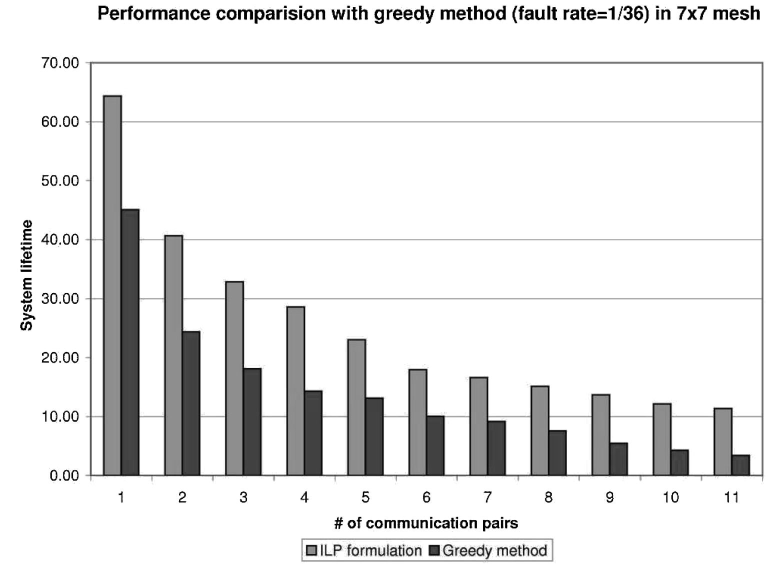

Fig. 18. Performance comparison of our technique with a greedy method ($\lambda_f = 1/36\, h^{-1}$).

as two independent design steps and each is quite complex. Therefore, optimizing the final solution or finding the near-optimal answer becomes even more challenging. Yet, our proposed formulation attempts to perform both two design steps simultaneously while acquiring a lower bound on the solution.

—Power consumption and reliability are two major performance measures in embedded computing systems. Given that the aforementioned objective metrics have different nature, optimizing both is impossible. However, both affect the lifetime of the system independently. By applying the concept of "lifetime" and defining "fault lifetime" and "power lifetime," these objectives are merged into a single measure. The resulting ILP algorithm guarantees to optimize system lifetime, taking into consideration both objectives. Usually in such scenarios, the problem is treated as a constrained optimization in which one objective is bounded (becomes a constraint) while the other is optimized. Our proposed ILP scheme uses the same concept in two different phases, which provably results in optimizing both objectives.

Our proposed ILP technique is an offline task in our system design procedure. In other words, it does not attempt to reconfigure the system after the design stage. The offline solution is used to optimally place and route sensors and actuators in grid-based networks. In this article, we are not considering any "recovery" scheme from faults. The main objective is to "avoid" faults by placing and routing the sensors using locations less prone to tears. Specifically, with respect to our target architecture and the nature of faults in corresponding applications, a tear can potentially cut many communication links at the same time whereas in fault-tolerant routing algorithms, recovery from single link loss (or a limited number of link loss) is being considered.

Maximizing system lifetime in more general network topologies may be considered for future works. Although a grid network is a common topology for many wearable embedded networked systems, removing this limitation can potentially lead to a more general methodology for fault-prone networks. In addition, exploiting network architectures in which recovery from faults can be done efficiently is unquestionably an interesting and a challenging problem that may be studied in the future.

REFERENCES

Barrenechea, G., Beferull-Lozano, B., and Vetterli, M. 2004. Lattice sensor networks: Capacity limits, optimal routing and robustness to failures. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*. ACM, New York, 186–195.

Bona, M. 1994. *Textile Quality*. Texilia, Manchester, UK.

Burstein, M. and Pelavin, R. 1983. Hierarchical wire routing. *IEEE Trans. Comput. Aid. Des. Integr. Circ. Syst. 2*, 4, 223–234.

Caldwell, A. E., Choi, H.-J., Kahng, A. B., Mantik, S., Potkonjak, M., Qu, G., and Wong, J. L. 1999. Effective iterative techniques for fingerprinting design ip. In *Proceedings of the 36th ACM/IEEE Conference on Design Automation (DAC'99)*. ACM, New York, 843–848.

Chan, T. F., Cong, J., Kong, T., and Shinnerl, J. R. 2000. Multi-level optimization for large-scale circuit placement. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'00)*. IEEE, Los Alamitos, 171–176.

Dunlop, A. E. and Kernighan, B. W. 1985. A Procedure for Placement of Standard Cell VLSI Circuits. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. 4*, 1, 92–98.

Hall, K. 1970. An r-dimensional quadratic placement algorithm. *Manage. Sci. 17,* 3, 219–229.

Kao, J.-C. and Marculescu, R. 2005. Energy-aware routing for e-textile applications. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE'05)*. IEEE Computer Society, Washington, DC, 184–189.

Karypis, G., Aggarwal, R., Kumar, V., and Shekhar, S. 1997. Multi-level hyper-graph partitioning: Application in vlsi domain. In *Proceedings of the 34th Annual Conference on Design Automation (DAC'97)*. ACM, New York, 526–529.

KLEINHANS, J. M., SIGL, G., JOHANNES, F. M., AND ANTREICH, K. J. 1991. Gordon: Vlsi Placement by quadratic programming and slicing optimization. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. 10*, 3, 365.

LEE, J. 1961. An algorithm for path connection and its application. *IEEE Trans. Electron. Comput. EC-10*, 3, 346–365.

MEOLI, D. AND MAY-PLUMLEE, T. 2002. Interactive electronic textile development: A review of technologies. *J. Textile Apparel Tech. Manage. 2*, 2, 1–12.

N. LEK, R. T. AND KANG, S. 1992. A new global router using zero-one integer linear programming techniques for sea-of-gates and custom logic arrays. *IEEE Trans. Comput.-Aid Des. Integr. Circ. Syst. 11*, 12, 1479–1494.

NI, L. M. AND MCKINLEY, P. K. 1993. A survey of wormhole routing techniques in direct networks. *IEEE Comput. 26*, 2.

PRIMENTAS, A. 2001. Puncture and tear of woven fabrics. *J. Textile Apparel Tech. Manage. 1*, 4, 1–8.

QUINN, N. AND BREUER, M. 1979. A force directed component placement procedure for printed circuit boards. *IEEE Trans. Circ. Syst. 26*, 6, 377–388.

SARRAFZADEH, M. AND WONG, C. K. 1996. *An Introduction to VLSI Physical Design*. McGraw- Hill Higher Education, New York.

SHAHOOKAR, K. AND MAZUMDER, P. 1990. A genetic approach to standard cell placement using meta-genetic parameter optimization. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. 9*, 5, 500–511.

SHERWANI, N. A. 1995. *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers, Norwell, MA.

SHRAGOWITZ, E. AND KEEL, S. 1987. A global router based on a multi-commodity flow model. *Integr. VLSI J. 5*, 1, 3–16.

STANLEY-MARBELL, P., MARCULESCU, D., MARCULESCU, R., AND KHOSLA, P. K. 2003. Modeling, analysis, and self-management of electronic textiles. *IEEE Trans. Comput. 52*, 8, 996–1010.

SUN, W. J. AND SECHEN, C. 1995. Efficient and effective placement for very large circuits. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. 14*, 3, 349–359.

WANG, M., YANG, X., AND SARRAFZADEHF, M. 2000. Dragon2000: Standard-cell placement tool for large industry circuits. In *Proceedings of the International Conference on Computer-Aided Design (ICCASD'00)*. IEEE Computer Society, Los Alamitos, 260–263.

WITKOWSKA, B. AND FRYDRYCH, I. 2004. A comparative analysis of tear strength methods. *Fibres and Textiles in Eastern Europe 12*, 2, 42–47.

Received August 2005; revised March 2006, August 2006; accepted August 2006